

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”  
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра информационных технологий управления

Платформа для создания и управления портфолио «Artfolio»

Курсовой проект по дисциплине  
«Технологии программирования»

09.03.02 Информационные системы и технологии  
Информационные системы и технологии управления предприятием

Зав. кафедрой	_____ д.т.н., профессор М. Г. Матвеев
Обучающийся	_____ ст. 3 курса оч. отд. И. В. Балык
Обучающийся	_____ ст. 3 курса оч. отд. Р. Р. Джаббаров
Обучающийся	_____ ст. 3 курса оч. отд. М. М. Сторожук
Руководитель	_____ В. А. Ушаков, преподаватель
Руководитель	_____ В.С. Тарасов, ст. преподаватель _____.20__

Воронеж 2024

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Постановка задачи.....	5
1.1 Требования к разрабатываемой системе .....	5
1.1.1 Функциональные требования .....	5
1.1.2 Нефункциональные требования .....	7
2 Анализ предметной области .....	8
2.1 Терминология (гlossарий) предметной области .....	8
2.2 Обзор аналогов .....	10
2.2.1 Behance.....	10
2.2.2 Coroflot .....	15
2.2.3 Dribbble .....	18
3 Диаграммы .....	22
3.1 Диаграмма прецедентов (Use-case diagram).....	22
3.2 Диаграмма деятельности (Activity diagram).....	26
3.3 Диаграмма последовательности (Sequence diagram).....	27
3.4 Диаграмма развёртывания (Deployment diagram).....	32
3.5 Диаграмма состояний (Statechart diagram) .....	32
3.6 ER–диаграмма .....	34
4 Реализация.....	35
4.1 Средства реализации.....	35
4.2 Разработка frontend (клиентской части веб-приложения) .....	37
4.3 Разработка backend (серверной части веб-приложения).....	39
4.4 Основная функциональность сайта.....	41
4.4.1 Для пользователя .....	41
4.4.2 Панель администратора .....	47

5 Тестирование .....	48
5.1 UI-тестирование .....	48
5.2 Дымовое тестирование .....	50
6 Анализ веб-приложения .....	52
ЗАКЛЮЧЕНИЕ .....	54

## **ВВЕДЕНИЕ**

В современном цифровом мире онлайн-платформы становятся неотъемлемой частью жизни в различных сферах, включая сферу творчества и дизайна. На смену традиционным методам продвижения и демонстрации своих работ приходят удобные, функциональные и доступные веб-платформы, предоставляющие уникальные возможности для создания и управления портфолио. Создание и управление портфолио через интернет не только упрощает процесс демонстрации работ, но и открывает новые возможности для продвижения и взаимодействия с потенциальными клиентами и аудиторией.

На протяжении последних лет спрос на онлайн-платформы для создания портфолио в данной сфере значительно возрос. Стремительное развитие технологий, доступность интернета и увеличение числа онлайн-аудиторий создают благоприятную среду для такого вида сервисов.

В современном цифровом мире, где онлайн-присутствие становится ключевым фактором успеха для художников, дизайнеров и фотографов, создание удобной и эффективной онлайн-платформы для управления портфолио представляет собой актуальную и необходимую задачу.

Цель данной курсовой работы состоит в разработке онлайн-платформы, предназначенной для художников, дизайнеров и фотографов. Она будет предоставлять пользователям возможность создания персонализированных портфолио, управления содержимым и взаимодействия с другими участниками.

## **1 Постановка задачи**

Целями данного курсового проекта является реализация системы, которая позволит:

- Пользователям загружать свои художественные работы, дизайны и фотографии, а также взаимодействовать с другими участниками, лайкая и комментируя их творчество;
- Художникам, дизайнерам и фотографам создавать портфолио и управлять ими, что способствует расширению их клиентской базы и профессиональной сети.

Целью данного курсового проекта является создание сайта, предназначенного для оформления продовольственных заказов с их доставкой пользователям по указанным адресам.

Сайт решает следующий ряд задач:

- Публикация творческих проектов, редактирование и удаление их;
- Получение информации о различных публикациях других пользователей;
- Оценивание публикаций других, возможность подписки на интересных пользователей.

### **1.1 Требования к разрабатываемой системе**

#### **1.1.1 Функциональные требования**

К разрабатываемому сайту выдвигаются следующие функциональные требования:

- при первом использовании сайта пользователь может зарегистрироваться и продолжить сеанс в авторизованной зоне;
- при повторном использовании сайта пользователь может авторизоваться в ранее созданный аккаунт;

- при повторном использовании сайта пользователь также может восстановить ранее созданный аккаунт в случае утери пароля;
- пользователь может листать публикации;
- пользователь может выбирать одну из публикаций для просмотра;
- пользователь может использовать поиск публикаций по названию;
- пользователь может выбирать критерии для получения публикаций в сетке;
- пользователь может просматривать данные профиля;
- пользователь может просматривать публикации профиля;
- пользователь может связаться с пользователем по электронной почте;
- пользователь может экспортировать описание профиля в формате PDF;
- авторизованный пользователь может подписываться на другие профили;
- авторизованный пользователь может изменять данные своего профиля;
- пользователь может просматривать подробную информацию о публикации;
- пользователь может читать комментарии к публикации;
- пользователь может листать фотографии в публикации;
- пользователь может перейти к автору публикации;
- авторизованный пользователь может ставить лайки к публикации;
- авторизованный пользователь может оставлять комментарии к публикации;
- авторизованный пользователь может удалять свои комментарии к публикации;
- авторизованный пользователь может подавать жалобы на комментарии и публикации других пользователей;
- администратор может удалять аккаунт пользователей;
- администратор может удалять комментарии к публикации;

- администратор может удалять публикацию;
- администратор может просматривать жалобы на комментарии и публикации;
- администратор может отмечать жалобу как рассмотренную.

### **1.1.2 Нефункциональные требования**

Сайт должен иметь архитектуру, соответствующую модели Клиент-Серверного взаимодействия на основе REST API.

Сайт должен быть оформлен в одной цветовой палитре с использованием ограниченного набора шрифтов. У страниц сайта должен быть единый стиль. В оформлении приложения должно присутствовать разработанное название.

Сайт и его страницы должны быть реализованы с поддержкой русской языковой версии.

Необходимо корректное и одинаковое отображение страниц сайта в следующих браузерах:

- Google Chrome 122.0.6261.95;
- Mozilla Firefox 123.0;
- Yandex Browser 24.1.1.944.

## **2 Анализ предметной области**

### **2.1 Терминология (гlossарий) предметной области**

Администратор – авторизованный пользователь с особыми привилегиями на сайте, ответственный за обеспечение соблюдения правил сайта. Его обязанности включают рассмотрение жалоб, удаление публикаций и комментариев, нарушающих правила сайта, а также удаление пользователей, которые многократно нарушают правила.

Пользователь – неавторизованный пользователь сайта, не имеющий полного доступа к функциям сайта.

Авторизованный пользователь – пользователь сайта, прошедший авторизацию, имеющий полный доступ к функциям сайта.

Аккаунт – учетная запись пользователя, которая позволяет войти в систему и получить доступ к определенным ресурсам и возможностям.

Профиль – веб-страница пользователя сайта с персональными данными о нем, а также с его публикациями.

Подписка – функция, которая позволяет авторизованным пользователям следить за публикациями других пользователей сайта.

Публикация – содержимое, представленное на сайте авторизованным пользователем, включающее в себя несколько фотографий работы пользователя, описание и название данной работы.

Автор публикации – это авторизованный пользователь, который создал и разместил данную публикацию на сайте.

Лайк – это отметка, которую авторизованный пользователь может оставить для выражения своего положительного отношения к публикации, размещённой на сайте. Лайки используются для измерения популярности публикаций и служат обратной связью для авторов.

Сайт (веб-приложение) – интернет-ресурс, состоящий из одной, нескольких или множества виртуальных страниц.



Сервер (серверная часть) – программно-аппаратная часть сервиса, которая хранится на сервере, обрабатывает полученные данные и отправляет ответ обратно.

Клиент (клиентская сторона) – интерфейс с набором функций, с которым взаимодействует пользователь.

## **2.2 Обзор аналогов**

Прежде, чем начинать разработку сайта для создания и управления портфолио, необходимо проанализировать уже имеющиеся платформы для понимания их плюсов и минусов. После сбора информации можно переходить к этапу разработки с учетом сделанных выводов по изученным аналогам.

### **2.2.1 Behance**

Behance — популярная среди дизайнеров и иллюстраторов всего мира социальная медиа-платформа, принадлежащая Adobe. Она позволяет пользователям размещать свои творческие работы, а также просматривать и оценивать чужие работы.

После входа на сайт пользователь видит главный экран с сеткой популярных публикаций. В блоке публикаций расположена кнопка «Фильтр», позволяющая выбрать критерии для показа публикаций в сетке, а также форма поиска. Пользователь может просматривать название публикации, количество лайков и просмотров, фото обложки публикации. Также пользователь может выбрать заинтересовавшую его публикацию и открыть её. На рисунке 1 показана главная страница сайта Behance до авторизации в системе, содержащая сетку публикаций.

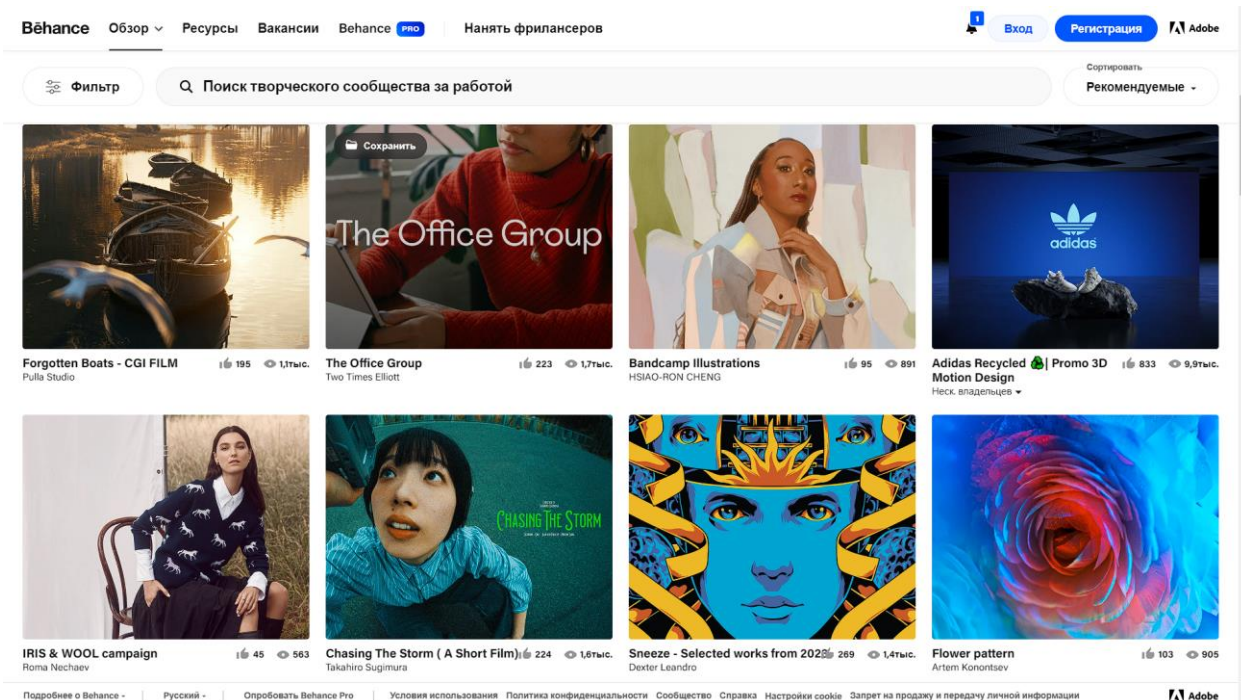


Рисунок 1 - Главная страница сайта «Behance»

Публикация открывается как модальное окно поверх основного содержимого. Вверху окна публикации расположен аватар автора публикации, название публикации, и имя автора. При нажатии на аватар и имя можно перейти на страницу автора. Ниже идут фотографии публикации. В низу окна расположен блок комментариев. Оставлять комментарии можно только после авторизации в системе. На рисунках 2-3 показана страница публикации сайта Behance.

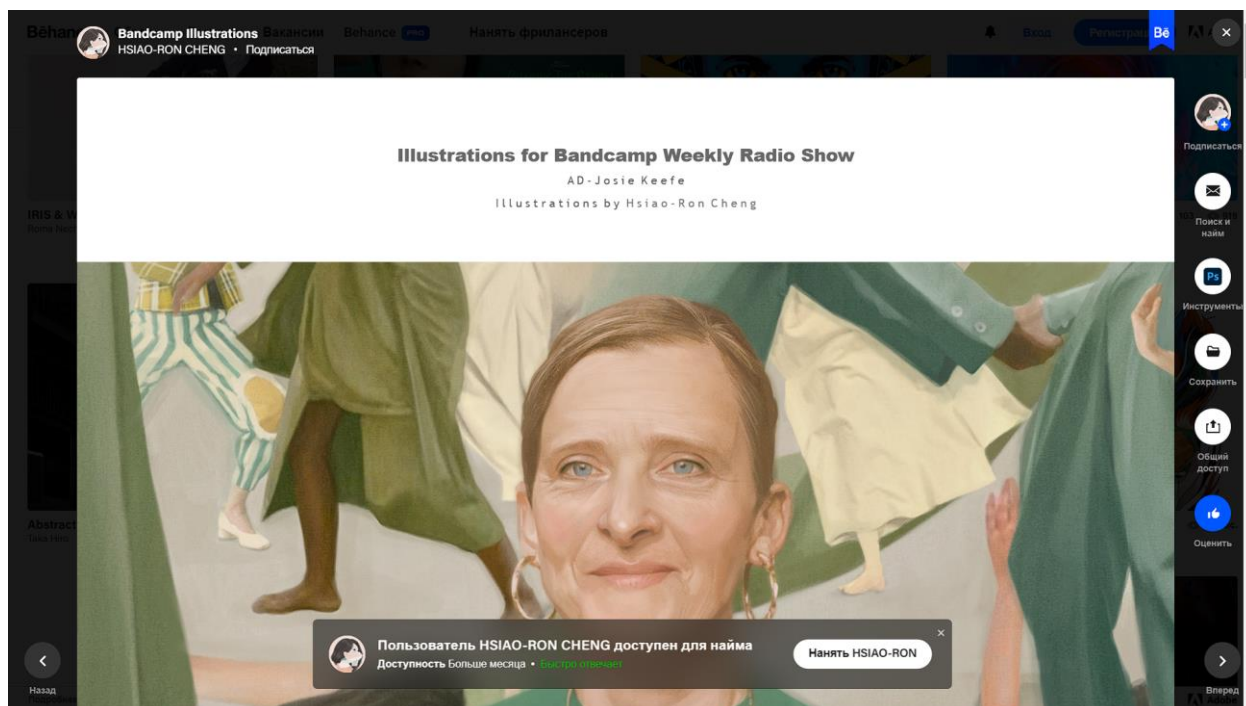


Рисунок 2 - Страница публикации сайта «Behance»

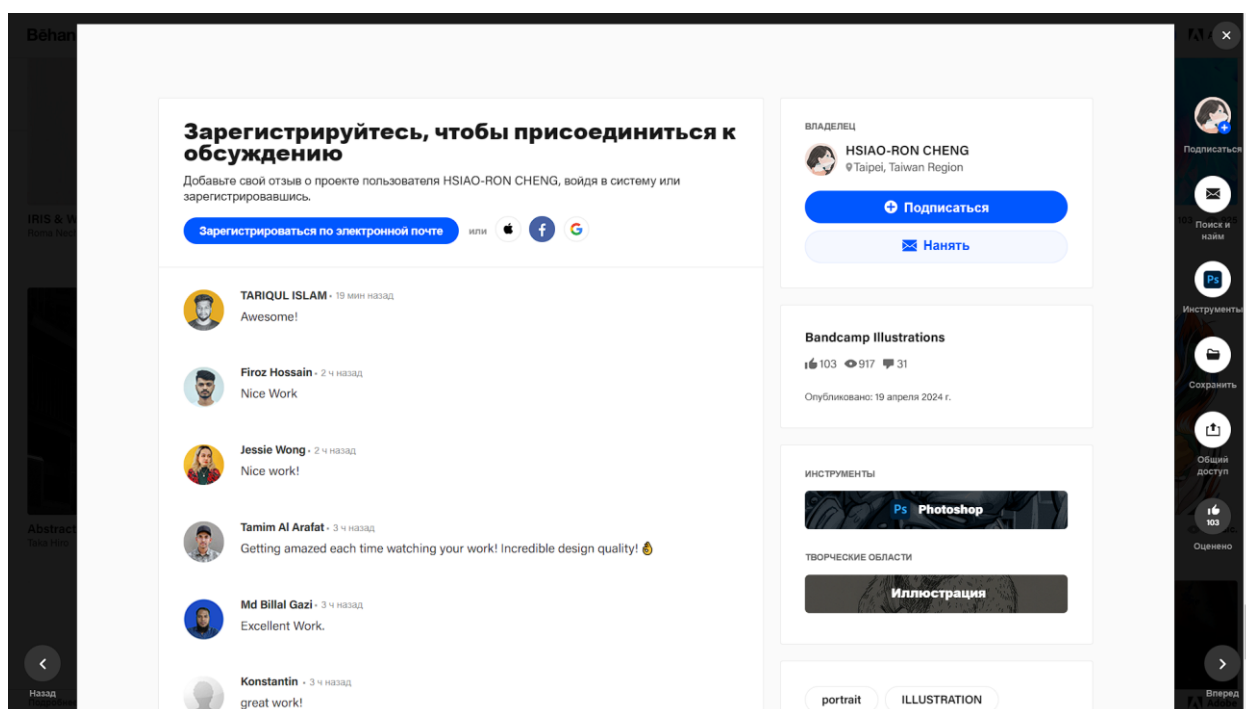


Рисунок 3 - Страница публикации сайта «Behance»

После авторизации появляется возможность комментировать публикации пользователей, подписываться на них. На рисунке 4 показана главная страница сайта Behance после авторизации в системе.

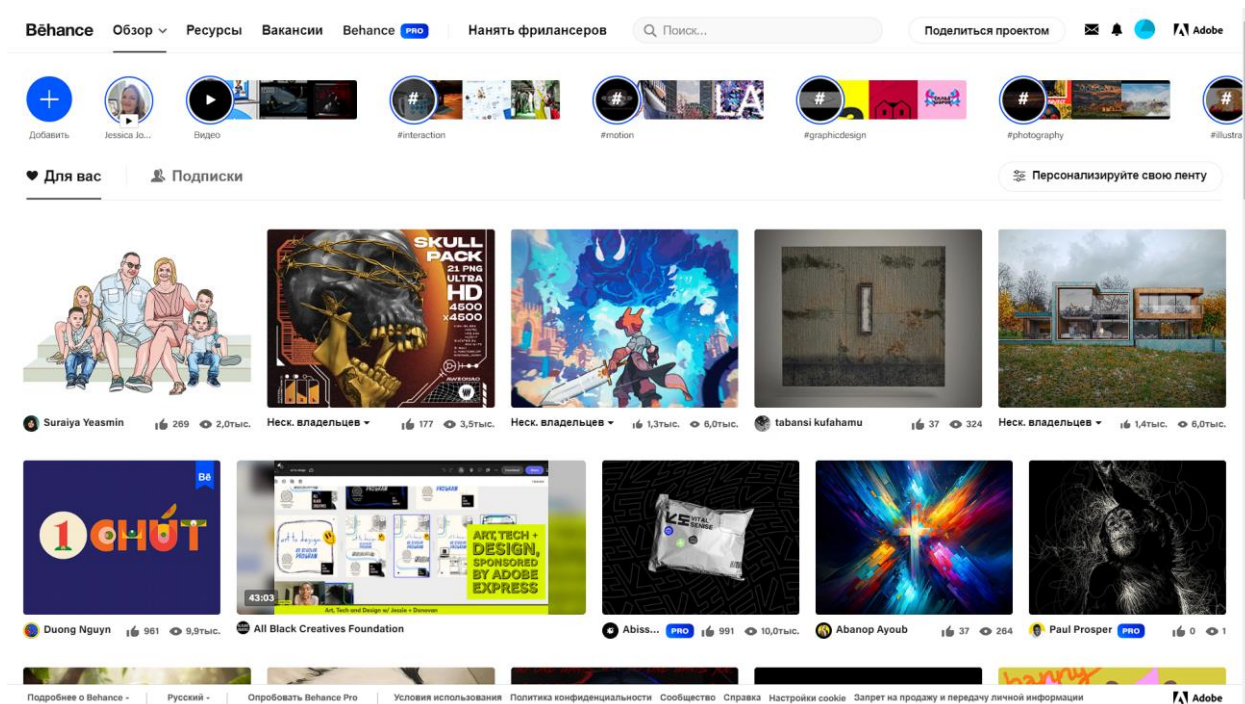


Рисунок 4 - Главная страница сайта «Behance» после авторизации

Для просмотра аккаунта можно нажать кнопку в шапке сайта «Аккаунт». Отобразится страница с данными аккаунта, с возможностью редактировать профиль.

Преимущества сайта:

- Сайт позволяет просматривать и оценивать творческие работы других людей, а также выкладывать свои;
- Возможность подписываться на других людей и связываться с ними;
- Возможность комментирования работ других пользователей;
- Возможность просмотра вакансий или свободных фрилансеров для найма.

Недостатки сайта:

- Достаточно сложный и перегруженный интерфейс;

- Отсутствие возможности экспортировать описание интересующего профиля себе;
- Нет отдельного блока для описания публикации.



### 2.2.2 Coroflot

Coroflot – это социальная сеть для дизайнеров, где можно размещать свои публикации, отслеживать вакансии и размещать резюме. Сайт позволяет пользователям выкладывать свои публикации, просматривать и оценивать публикации других людей. Также есть возможность просмотра вакансий. Для просмотра свободных фрилансеров необходимо приобрести платную подписку.

После входа на сайт пользователь видит главный экран с демонстрацией основных возможностей сайта. На рисунке 5 показана главная страница сайта Coroflot. Для просмотра работ других пользователей необходимо нажать кнопку «Discover» в навигационной панели. На рисунке 6 показана страница, содержащая работы других пользователей.

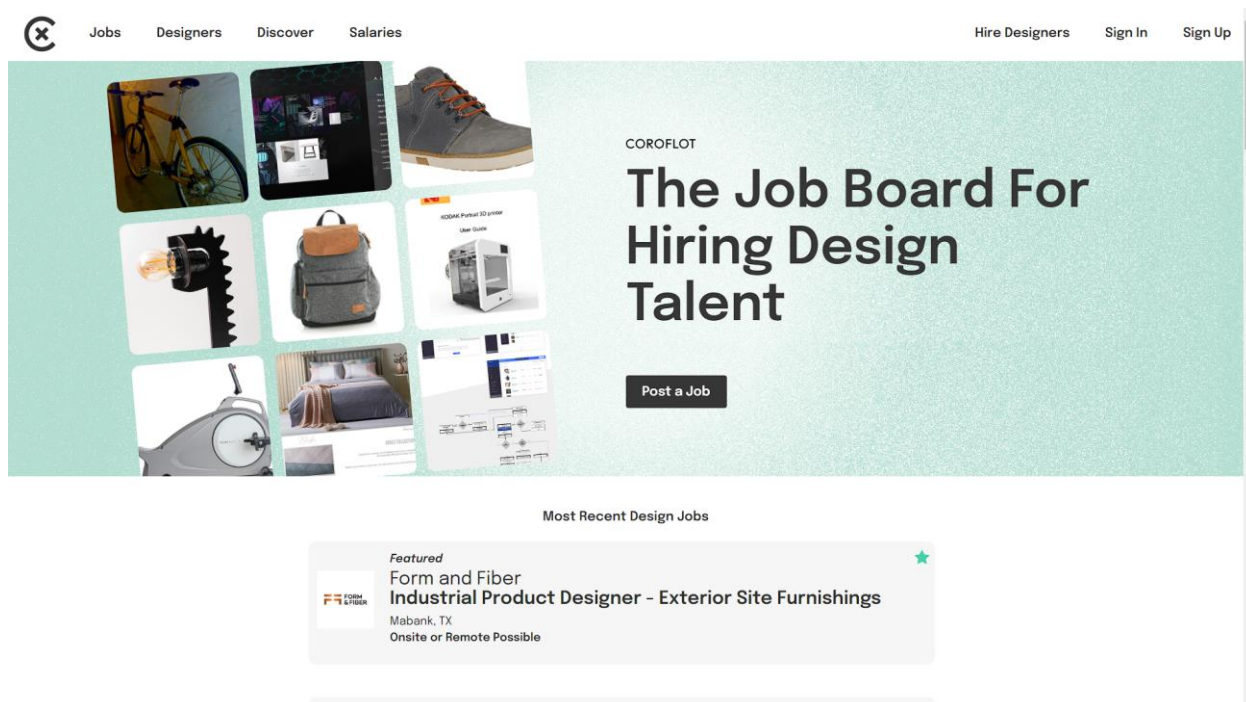


Рисунок 5 - Главная страница сайта «Coroflot»

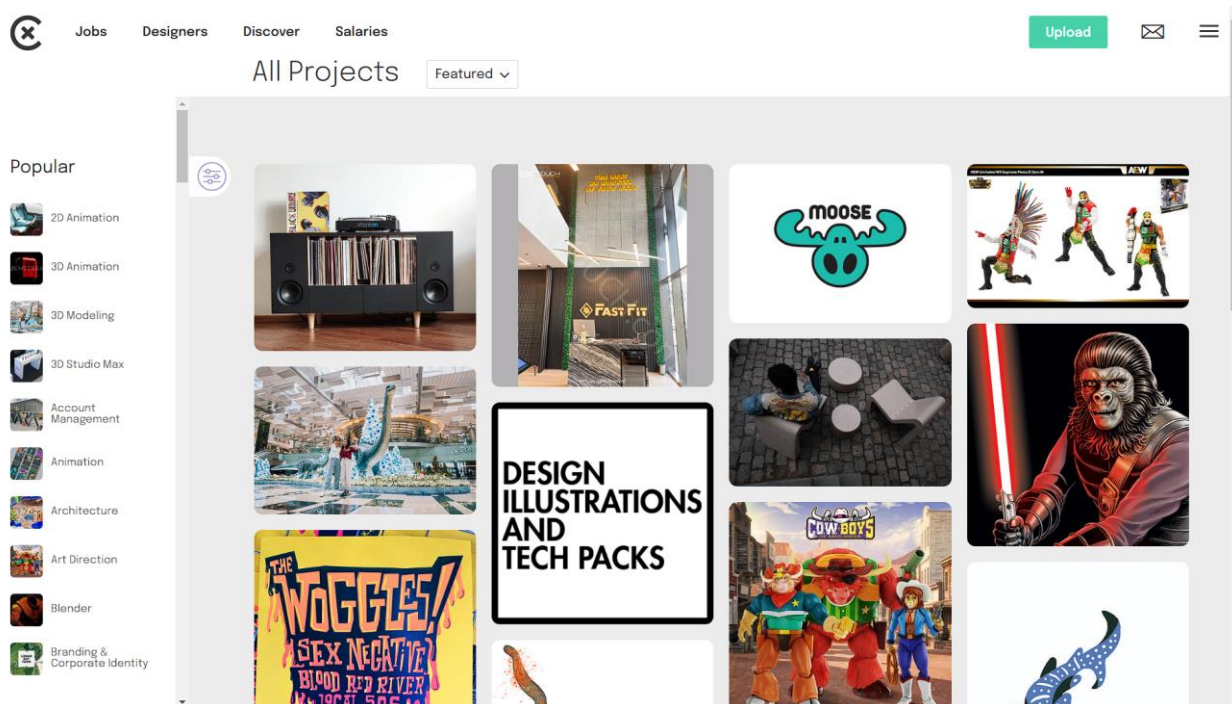


Рисунок 6 - Страница публикаций сайта «Coroflot»

Для просмотра конкретной публикации можно нажать на неё. Вверху отображается аватар и имя автора. Ниже идут фотографии публикации. Под ними название публикации, описание публикации и комментарии. Оставлять комментарии могут только зарегистрированные пользователи, прошедшие верификацию. На рисунках 7-8 показана страница публикации сайта Coroflot.





Рисунок 7 - Страница публикации сайта «Coroflot»

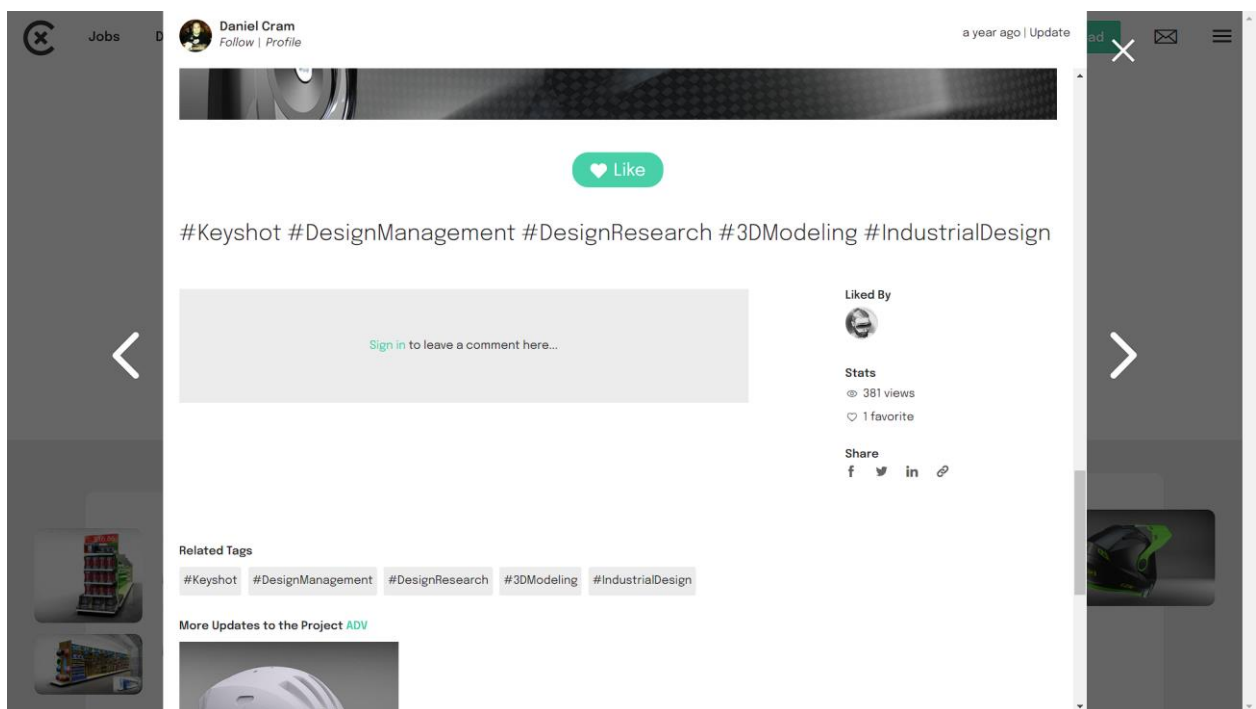


Рисунок 8 - Страница публикации сайта «Coroflot»

Преимущества сайта:

- Сайт позволяет просматривать и оценивать творческие работы других людей, а также выкладывать свои;
- Возможность подписываться на других людей и связываться с ними;
- Возможность комментирования работ других пользователей;
- Возможность просмотра вакансий.

Недостатки сайта:

- Отсутствие возможности экспортировать описание интересующего профиля себе;
- Отсутствие русской языковой версии;
- Для поиска свободных фрилансеров необходимо приобрести платную подписку;
- Для оставления комментариев необходимо пройти верификацию.

### **2.2.3 Dribbble**

Dribbble — это платформа саморекламы и социальная сеть для творческих людей.

После входа на сайт пользователь видит главный экран с демонстрацией основных возможностей сайта. Ниже располагается лента с популярными публикациями. На рисунке 9 показана главная страница сайта Dribbble до авторизации в системе. После авторизации в системе пользователю предоставляется страница с сеткой из рекомендуемых публикаций. На ней можно выбрать интересующую сферу для выборки публикаций и другие критерии для сортировки и фильтрации. На рисунке 10 показана страница публикаций сайта Dribbble после авторизации в системе.

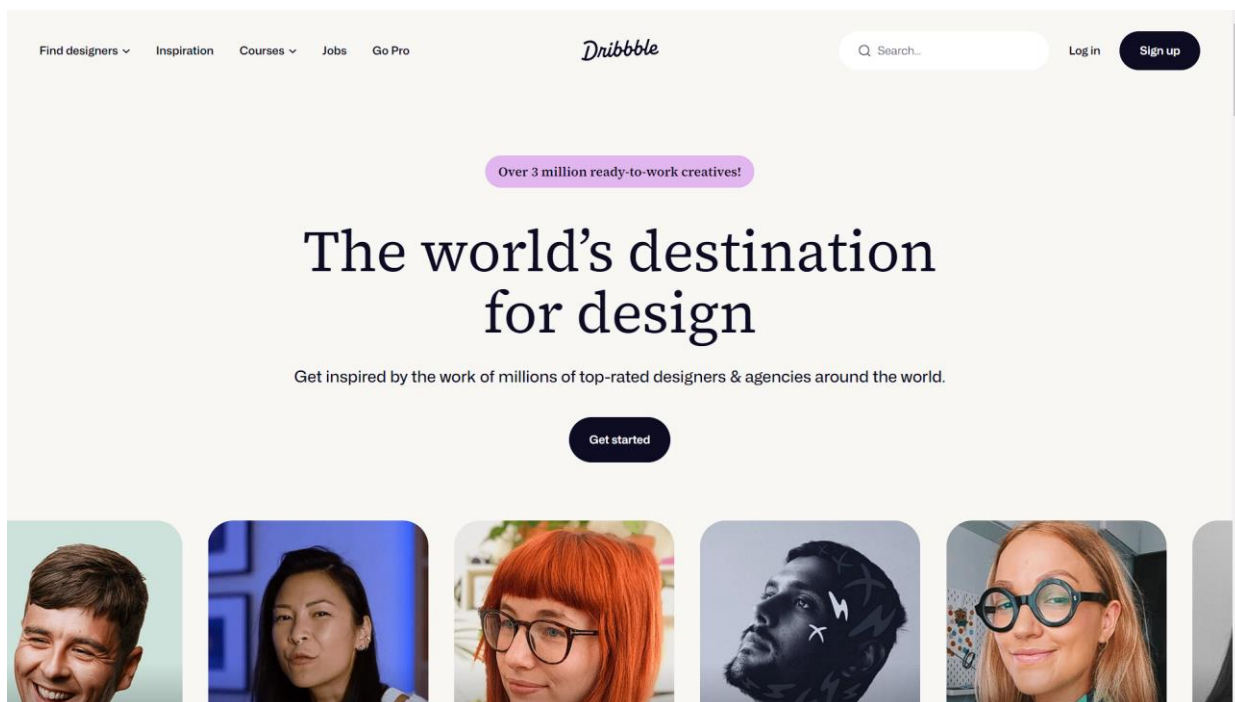


Рисунок 9 - Главная страница сайта «Dribbble»

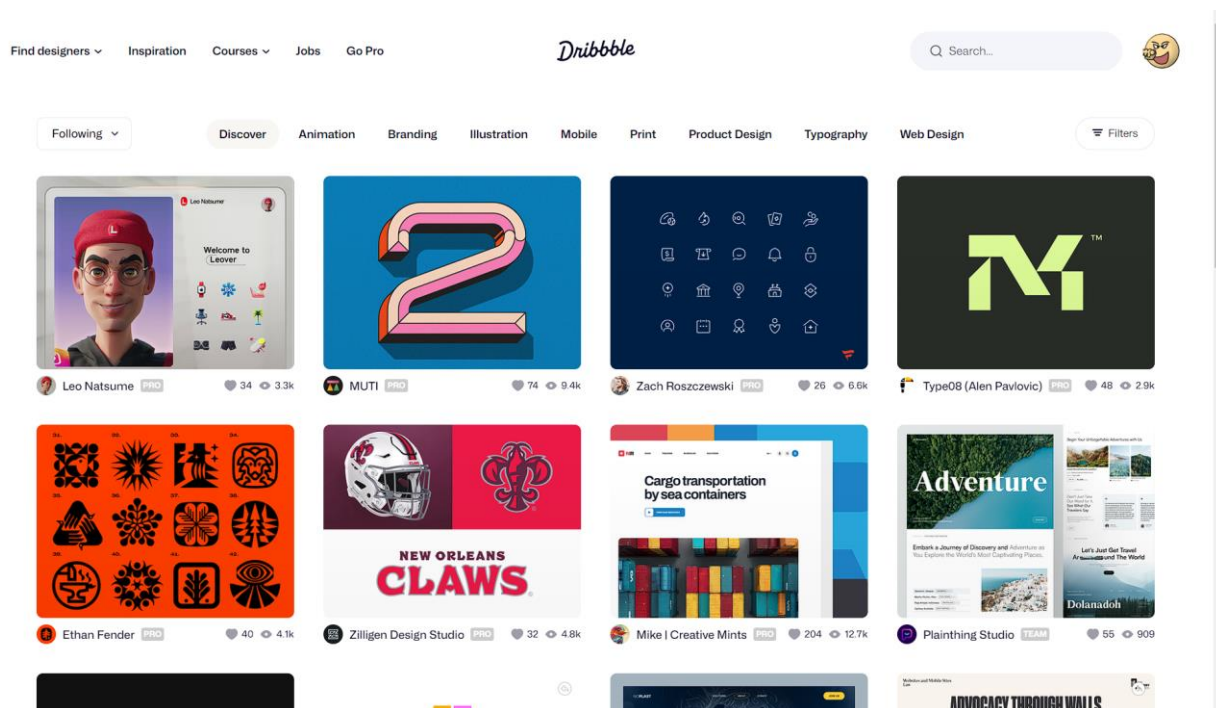


Рисунок 10 - Страница публикаций сайта «Dribbble»

Для просмотра конкретной публикации можно нажать на интересующую публикацию в сетке, после чего откроется модальное окно с публикацией. Вверху окна расположено название работы, аватар и имя автора публикации. Ниже располагаются сами фотографии работы. При нажатии на фотографию, откроется модальное окно с фотографией, где её можно приблизить и рассмотреть более детально. Справа расположены три кнопки: «Комментарии», «Поделиться» и «Подробнее». Комментировать публикации может авторизованный пользователь. Также, автор публикации может отключать возможность комментирования конкретной работы. На рисунке 11 показана страница публикации сайта Dribbble.

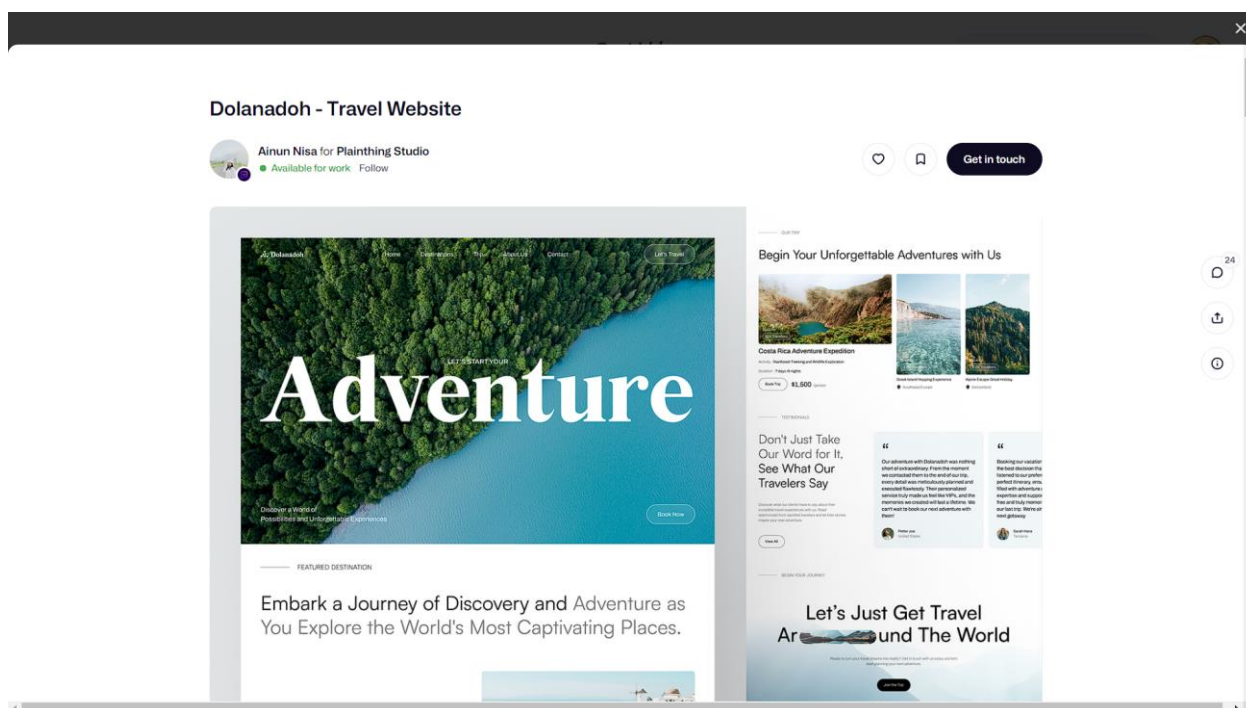


Рисунок 11 - Страница публикации сайта «Dribbble»

Добавить свою работу можно авторизованному пользователю после наведения на иконку профиля в навигационной панели.

Преимущества сайта:

- Сайт позволяет просматривать и оценивать творческие работы других людей, а также выкладывать свои;

- Возможность подписываться на других людей и связываться с ними;
- Возможность комментирования работ других пользователей;
- Сайт имеет простой и понятный интерфейс.

Недостатки сайта:

- Отсутствие русской языковой версии;
- Отсутствие возможности экспортировать описание интересующего профиля себе.

Проанализировав несколько платформ, можно заметить, что даже у таких популярных проектов имеются недостатки. В реализуемом проекте необходимо избежать данных недостатков путем упрощения пользовательского интерфейса и улучшения функционала.

## **3 Диаграммы**

### **3.1 Диаграмма прецедентов (Use-case diagram)**

Диаграмма прецедентов представляет собой диаграмму, которая моделирует функциональность системы, показывая ее взаимодействие с актерами, внешними сущностями, которые взаимодействуют с системой. Диаграмма прецедентов фокусируется на функциональных возможностях системы.

На рисунках 12-14 изображена диаграмма прецедентов системы сайта Artfolio.



Рисунок 12 - Use-case diagram (Неавторизованный пользователь)

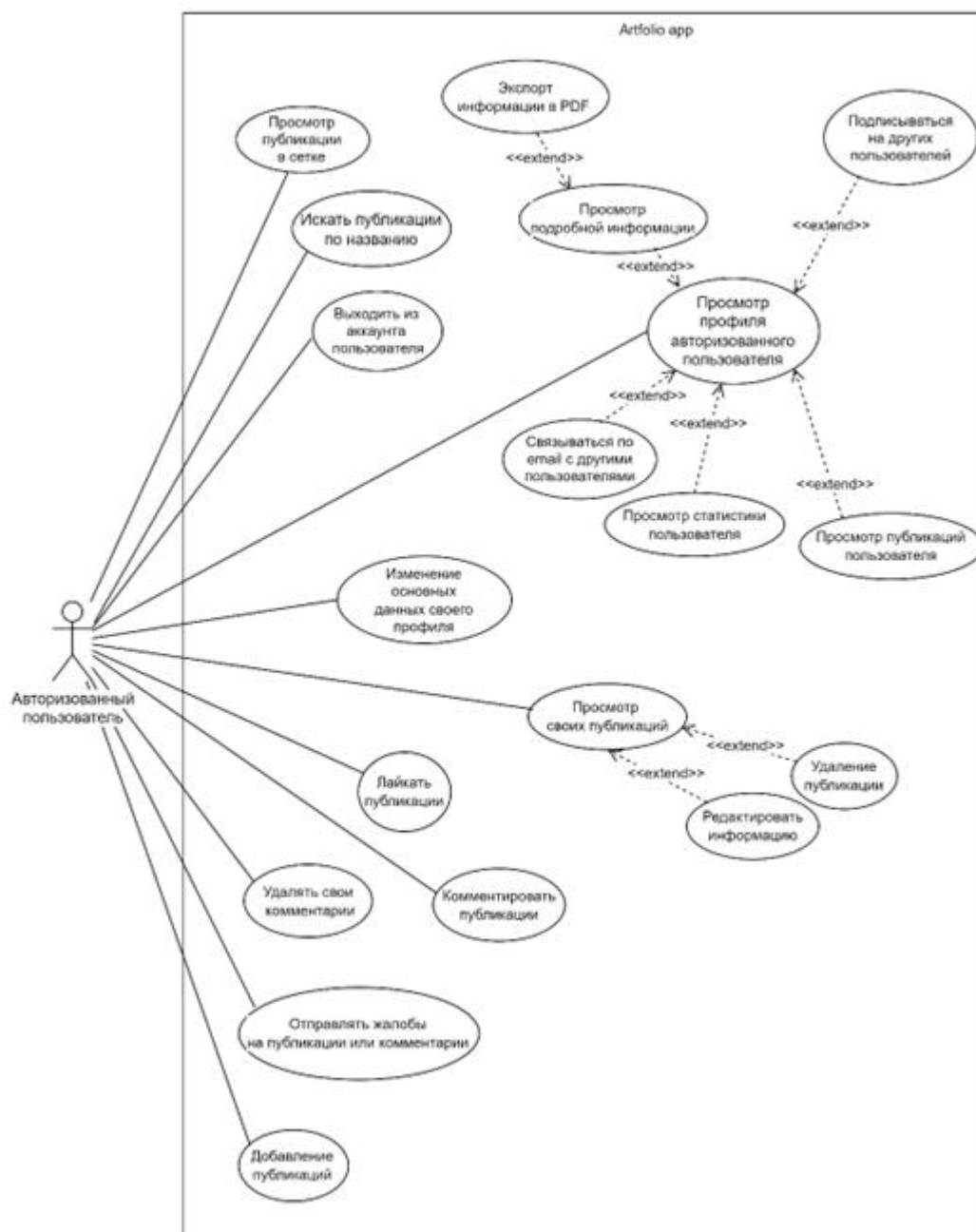


Рисунок 13 - Use-case diagram (Авторизованный пользователь)



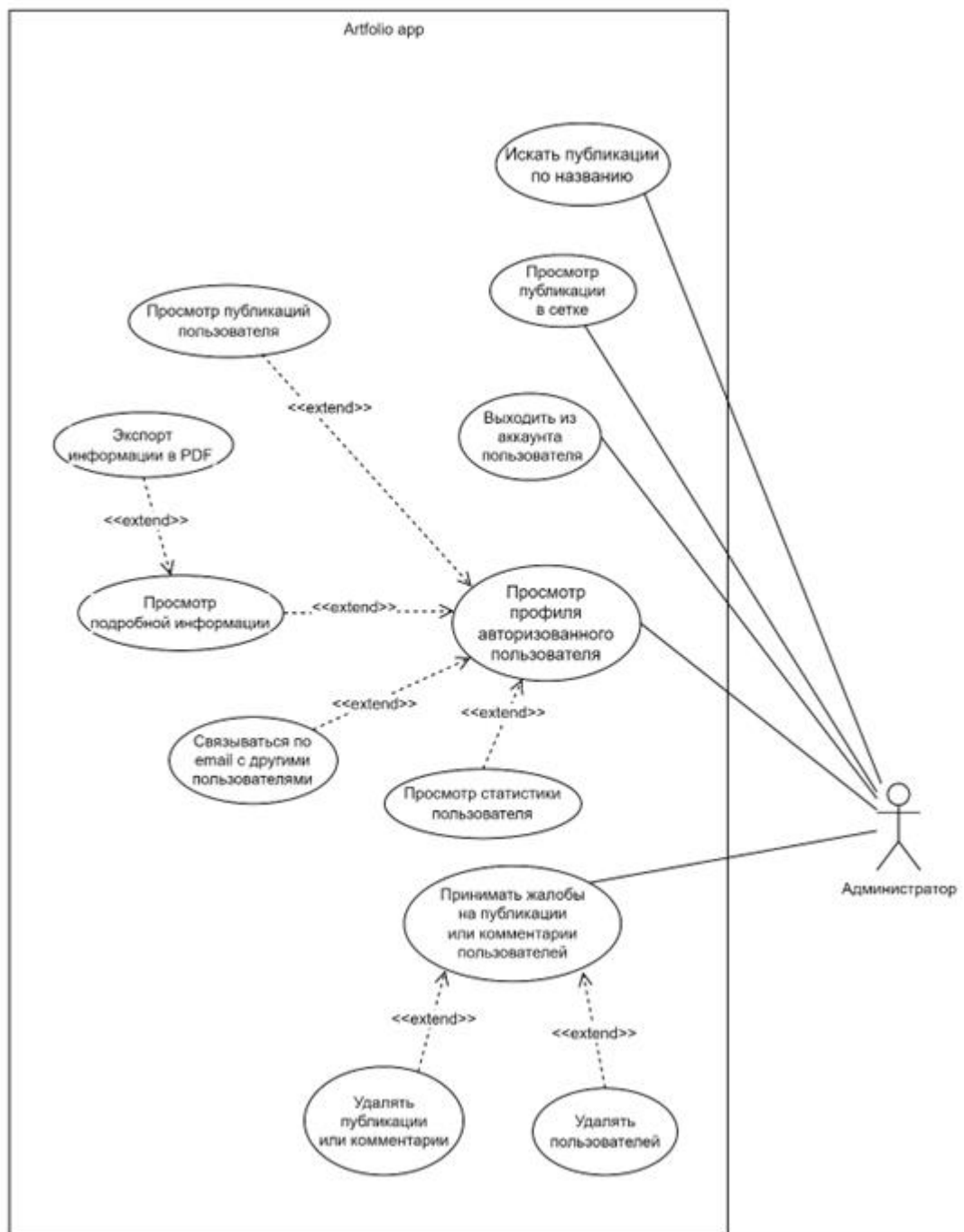


Рисунок 14 - Use-case diagram (Администратор)

### 3.2 Диаграмма деятельности (Activity diagram)

Диаграмма деятельности представляет собой графическую модель, которая позволяет описывать последовательность действий, процессы и поведение системы. Эта диаграмма используется для моделирования бизнес-процессов, алгоритмов и составных частей системы.

На рисунке 15 изображена диаграмма деятельности системы сайта Artfolio.

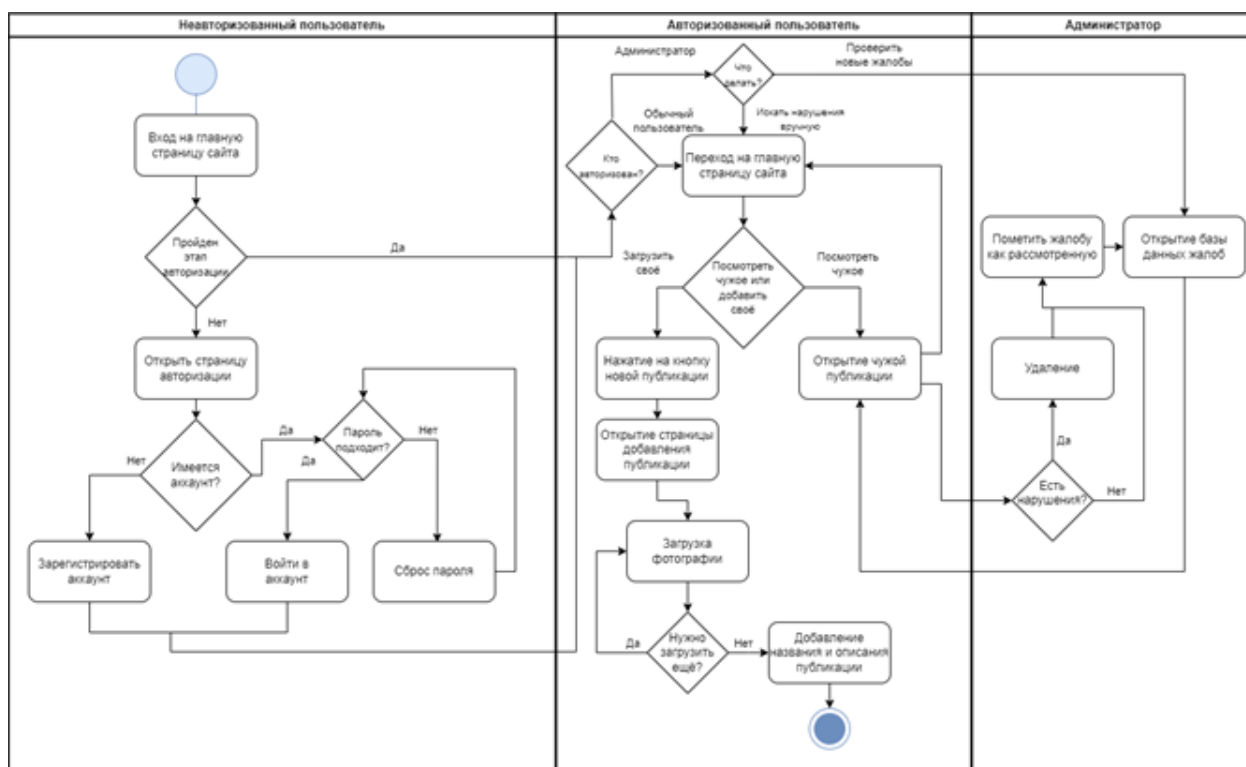


Рисунок 15 - Activity diagram

### **3.3 Диаграмма последовательности (Sequence diagram)**

Диаграмма последовательности представляет собой диаграмму, которая моделирует взаимодействие между объектами в определенной последовательности времени. Она позволяет показать, как объекты обмениваются сообщениями и взаимодействуют друг с другом для выполнения определенной функциональности.

На рисунках 16-19 изображена диаграмма последовательности системы сайта Artfolio.

# Диаграмма последовательностей

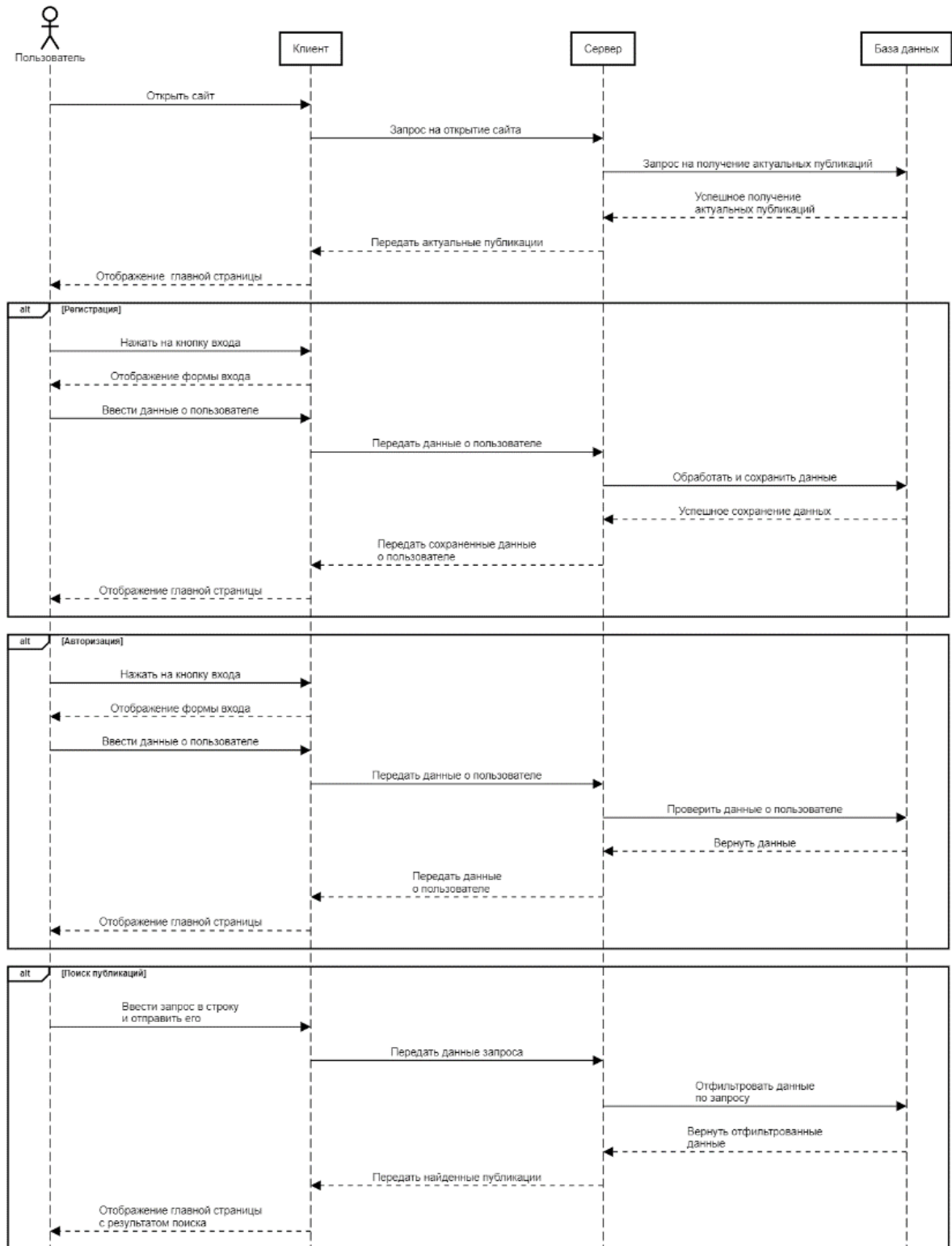


Рисунок 16 - Sequence diagram (1 часть)

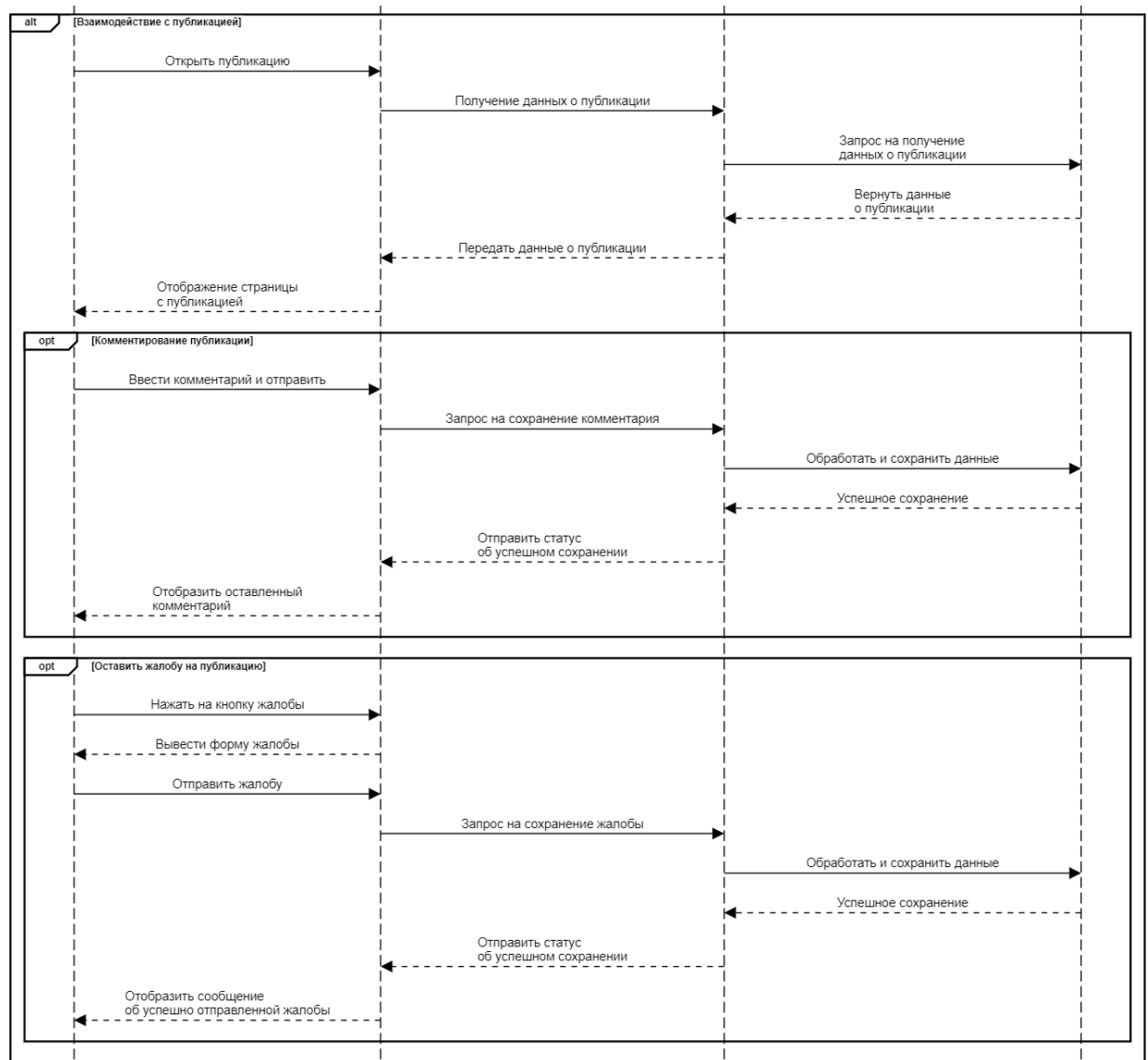


Рисунок 17 - Sequence diagram (2 часть)

# Диаграмма последовательностей

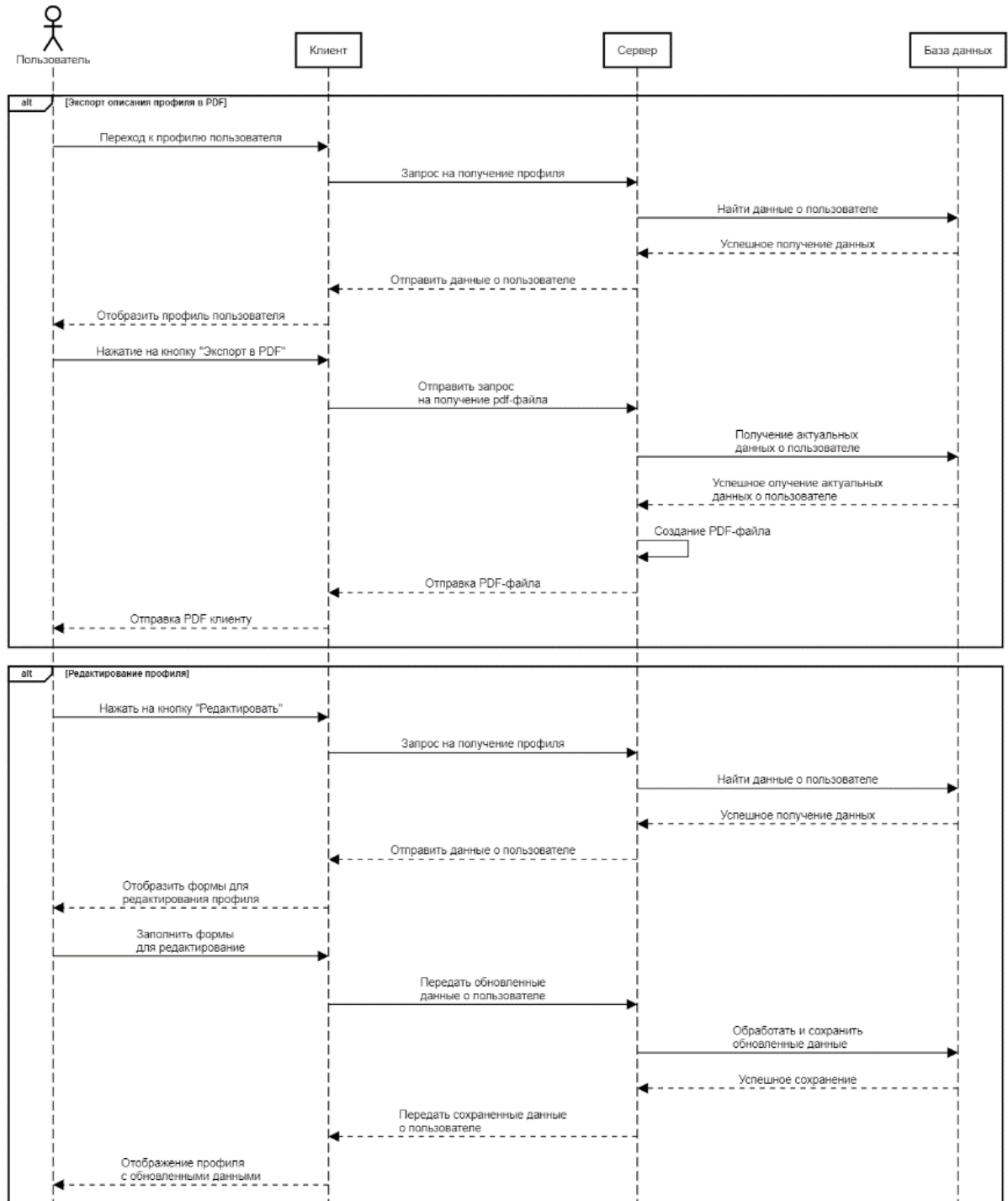


Рисунок 18 - Sequence diagram (3 часть)

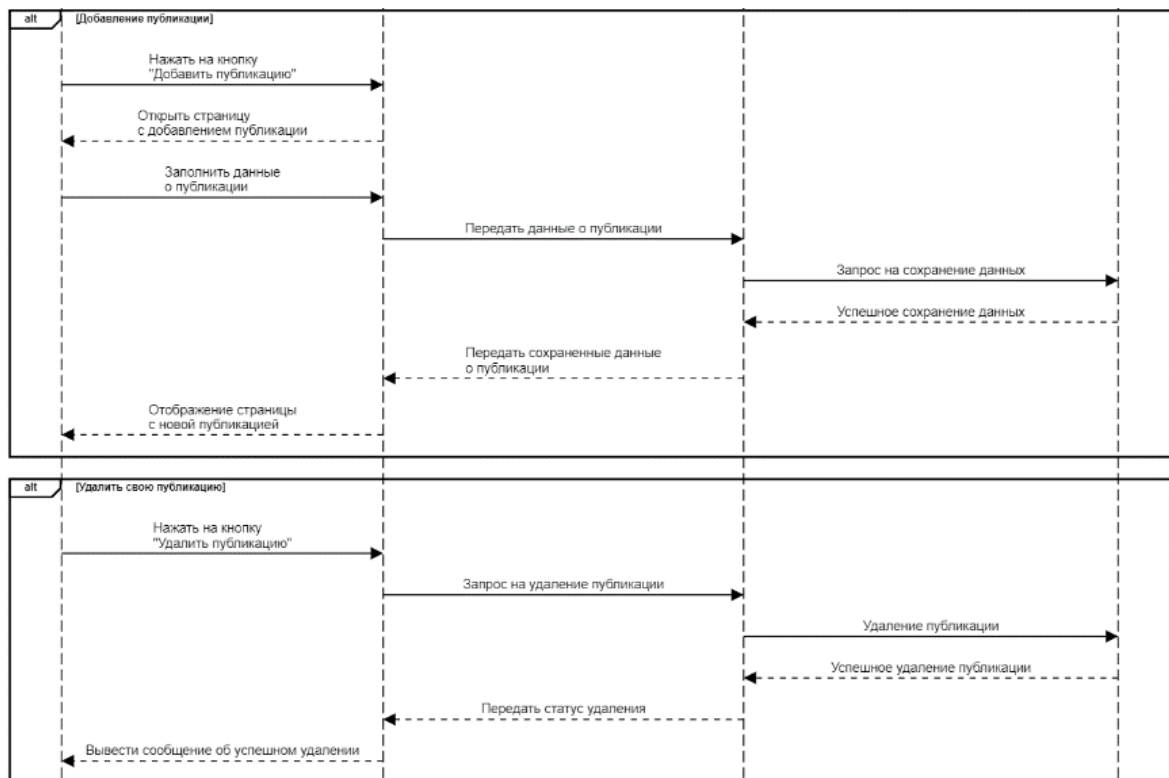


Рисунок 19 - Sequence diagram (4 часть)

### 3.4 Диаграмма развёртывания (Deployment diagram)

Диаграмма развёртывания представляет собой диаграмму, которая моделирует физическую конфигурацию аппаратного и программного обеспечения в рамках системы.

На рисунке 20 изображена диаграмма развёртывания системы сайта Artfolio.

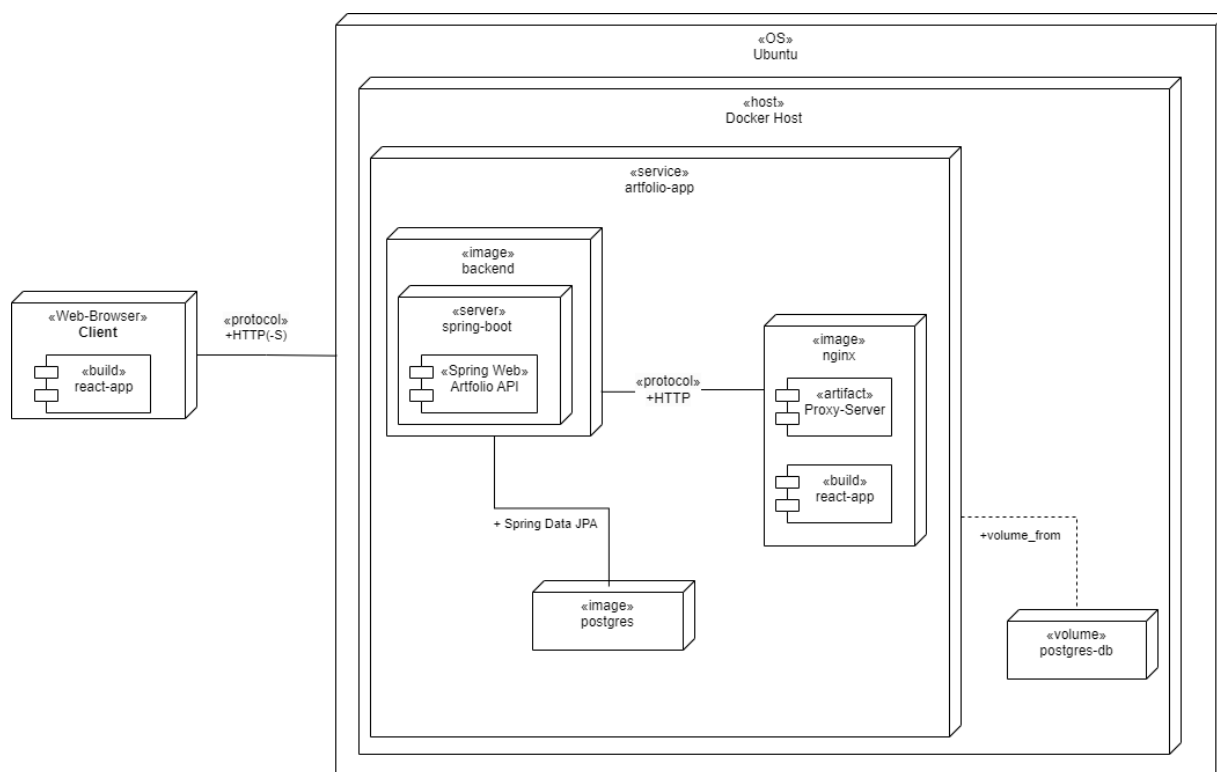


Рисунок 20 - Deployment diagram

### 3.5 Диаграмма состояний (Statechart diagram)

Диаграмма состояний представляет собой диаграмму, которая моделирует поведение объекта или системы в различных состояниях и переходах между этими состояниями. Она описывает, как объект или система реагирует на различные события, изменяя свое состояние.

На рисунке 21 изображена диаграмма состояния системы сайта Artfolio.



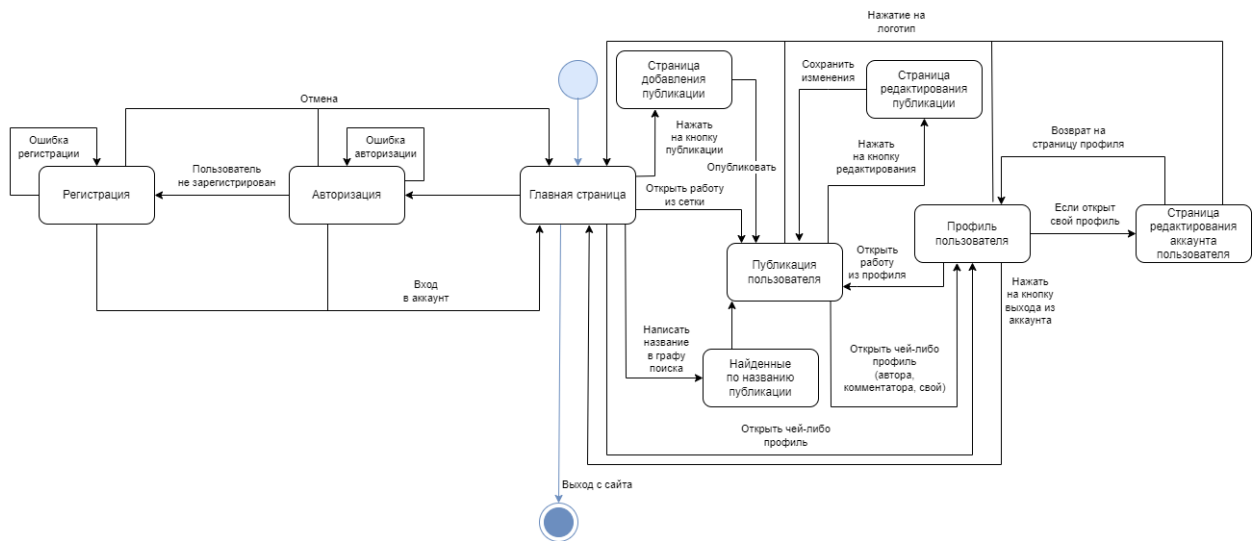


Рисунок 21 - Statechart diagram

### 3.6 ER-диаграмма

ER-диаграмма — это графическая модель, используемая в базах данных для описания сущностей и их взаимосвязей.

На рисунке 22 изображена ER-диаграмма системы сайта Artfolio.

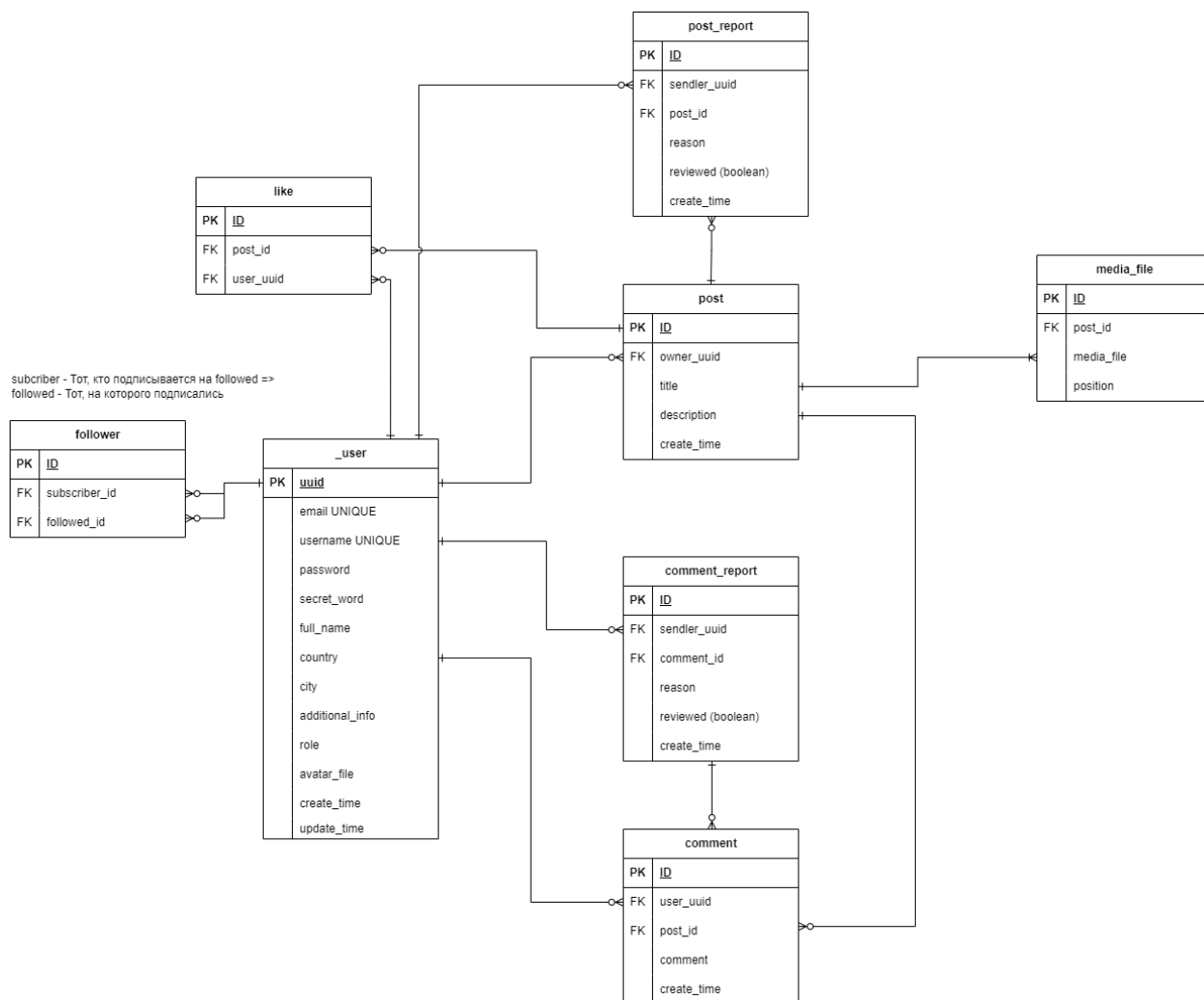


Рисунок 22 - ER-диаграмма

## 4 Реализация

### 4.1 Средства реализации

Веб-приложение использует клиент-серверную архитектуру и разделяется на две основные части: backend (сервер) и frontend (клиент). Обмен данными между ними осуществляется посредством REST API.

В процессе разработки был использован Git-flow для организации работы. В нашем проекте это модель ветвления включает две основные ветки: main и develop, а также временные ветки для добавления новых функций.

Для разработки серверной части сайта был выбран следующий ряд технологий:

- Java — многоплатформенный, объектно-ориентированный язык программирования, предназначенный для разработки различных приложений, включая мобильные, корпоративные, и серверные, обеспечивая быстроедействие, безопасность и надежность;
- Spring Framework — это фреймворк с открытым исходным кодом для языка программирования Java. Он был создан для упрощения разработки и поддержки масштабируемых, слабосвязанных и повторно используемых приложений;
- PostgreSQL — реляционная база данных с открытым кодом, которая поддерживается в течение 30 лет разработки и является одной из наиболее известных среди всех существующих реляционных баз данных;
- MinIO — это высокопроизводительная система объектного хранения с открытым кодом, которая совместима по API с облачным сервисом хранения Amazon S3.

Для разработки клиентской части сайта был выбран следующий ряд технологий:

- HTML — стандартизированный язык разметки, используемый для создания веб-страниц. Он определяет структуру и содержимое веб-страницы с помощью различных тегов и атрибутов. HTML состоит из элементов, которые обрамляют содержимое веб-страницы и определяют его семантику;
- CSS — это язык стилей, используемый для определения внешнего вида и форматирования веб-страниц, созданных с помощью HTML. CSS позволяет управлять цветами, шрифтами, размерами, расположением элементов и другими аспектами визуального представления веб-страницы;
- JavaScript — высокоуровневый язык программирования, который широко используется для разработки интерактивных веб-страниц и веб-приложений;
- TypeScript — это расширение языка JavaScript, которое обеспечивает строгую типизацию и используется для разработки клиентской части приложений.
- React — JavaScript-библиотека для создания пользовательских интерфейсов. Она позволяет разработчикам создавать масштабируемые и переиспользуемые компоненты, которые обновляются автоматически при изменении данных. React основан на концепции компонентного подхода, что делает код более организованным, понятным и легко поддерживаемым.

## 4.2 Разработка frontend (клиентской части веб-приложения)

Клиентская часть приложения была написана на языке TypeScript с использованием фреймворка React. Для отправки запросов с клиентской части приложения на серверную часть используется библиотека axios. Она предоставляет интерфейс для выполнения HTTP-запросов прямо из браузера.

Для создания модульной и переиспользуемой архитектуры, приложение было разбито на компоненты. Каждый компонент отвечает за конкретную функциональность или отображение определенной части пользовательского интерфейса.

Проект имеет следующую структуру (Рисунок 23):

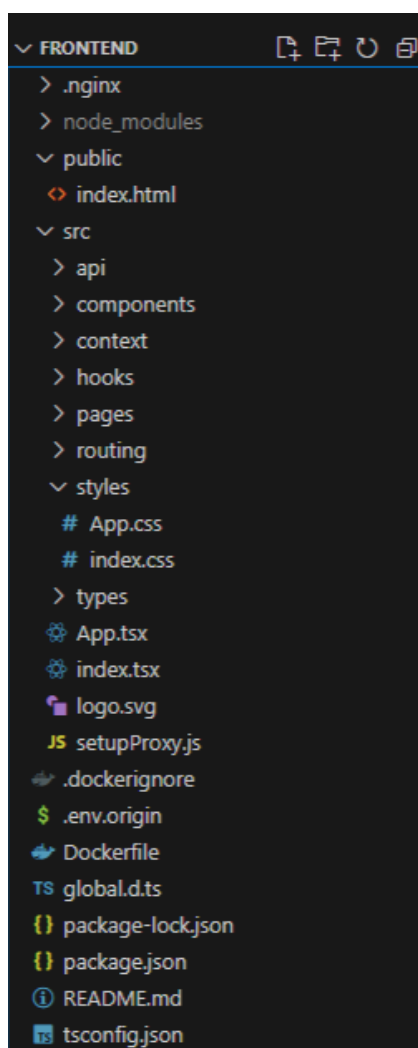


Рисунок 23 - Структура клиентской части приложения

- Модуль «api». Этот модуль содержит сервисы необходимые для связи с сервером, получением и отправкой данных;
- Модуль «components». Этот модуль содержит компоненты, которые используются в приложении;
- Модуль «context». Этот модуль содержит файлы, связанные с управлением контекстом и состоянием приложения;
- Модуль «hooks». Этот модуль содержит пользовательские функции, которые используются для управления состоянием и побочными эффектами в компонентах;
- Модуль «pages». Этот модуль содержит компоненты, представляющие отдельные страницы приложения;
- Модуль «routing». Этот модуль содержит файлы, связанные с маршрутизацией, такие как вспомогательные функции и константы маршрутов;
- Модуль «styles». Этот модуль содержит стили и файлы, связанные с оформлением приложения;
- Модуль «types». Этот модуль содержит определения типов и интерфейсов TypeScript, используемые в приложении;
- Файл «index.tsx». Этот файл является точкой входа в приложение. Он служит главным файлом, с которого начинается выполнение приложения;
- Файл «App.tsx». Этот файл является основным компонентом React, в который встраиваются остальные компоненты.

### 4.3 Разработка backend (серверной части веб-приложения)

Серверная часть приложения была написана на языке Java с использованием Spring Framework и СУБД PostgreSQL. Проект имеет следующую структуру (Рисунок 24):

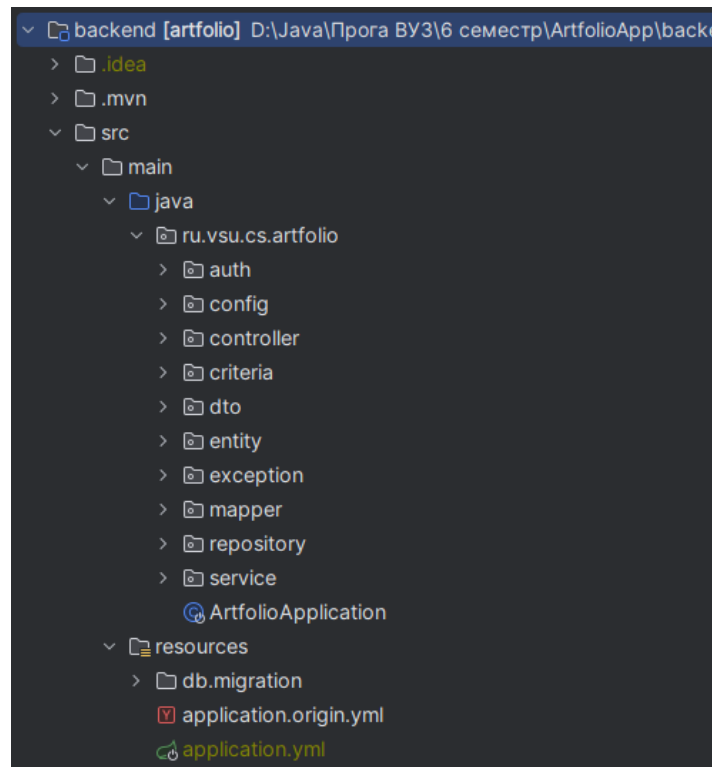


Рисунок 24 - Структура серверной части приложения

- Файл «ArtfolioApplication.java». Этот файл запускает весь контекст Spring, запускает сервер и подключается к базе данных, после чего запускает скрипты для миграций, находящиеся в «db/migration» пакете;
- Пакет «controller». Этот пакет содержит в себе функции и методы, ответственные за обработку запросов от клиентской части приложения. Внутри него находятся контроллеры, каждый из которых отвечает за определенные маршруты (endpoints) и передает соответствующие запросы к сервисам;

- Пакет «service». Этот пакет содержит в себе классы, которые отвечают за определенную бизнес-логику. Обращается к слою доступа к данным (repository);
- Пакет «repository». Этот пакет содержит в себе интерфейсы для прямой работы с базой данных. Такие операции как чтение, запись, удаление, обновления, фильтрация выполняются с помощью данных интерфейсов;
- Пакет «entity». Этот пакет содержит в себе модели данных, которые отображают сущности приложения и описывает их связь;
- Пакет «exception». Этот пакет содержит в себе классы исключений, которые приложение может выбросить, а также перехватчик таких исключений, который их обрабатывает и возвращает в качестве ответа клиенту;
- Пакет «auth». Этот пакет содержит в себе классы отвечающие за авторизацию;
- Пакет «config». Этот пакет содержит всю конфигурацию Spring boot приложения. Отвечает за создания необходимых объектов безопасности, связей с базой данных;
- Пакет «criteria». Содержит в себе статические функции, создающие динамически запросы к базе данных;
- Пакет «dto». Содержит в себе модели запросов и ответов клиенту. В них отображаются все сущности базы данных;
- Пакет «mapper». Содержит в себе классы, отвечающие за конвертацию сущностей в dto, а также dto в сущности.

Для развертывания приложения был использован виртуальный сервер (VPS) от провайдера Рег.ру. В качестве веб-сервера был выбран Nginx. Приложение было развернуто на этом сервере, а Nginx использован для обслуживания статического контента и проксирования запросов к



приложению. Для разворачивания был использован Docker — инструмент для контейнеризации приложений.

Для описания спецификации API использовался Swagger. Для интеграции его в проект потребовалось добавить библиотеку «springdoc-openapi-starter-webmvc-ui».

## 4.4 Основная функциональность сайта

### 4.4.1 Для пользователя

При входе на сайт пользователю открывается главная страница сайта с публикациями пользователей. На рисунке 25 показана главная страница сайта Artfolio до авторизации в системе.

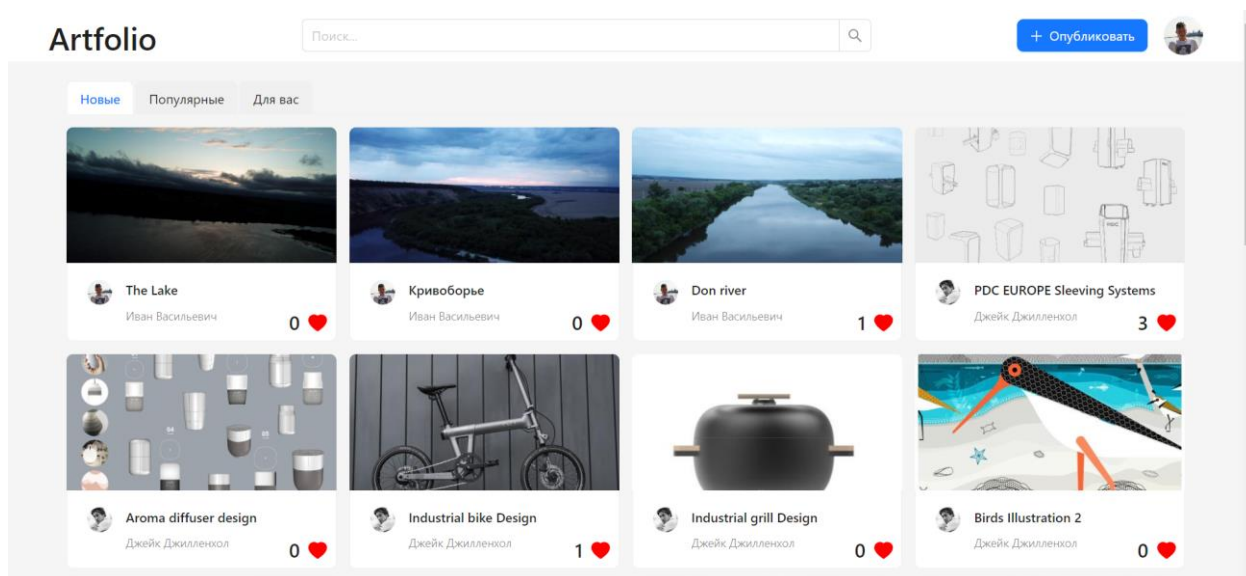


Рисунок 25 - Главная страница

На этой странице пользователь может выбрать интересующую публикацию или найти такую, используя строку поиска. Кроме этого, пользователь может загрузить собственную работу или авторизоваться, нажав на соответствующую кнопку в шапке сайта.

После выбора нужной публикации пользователь переходит на её страницу, где он, если это его публикация, может её редактировать, удалить или пожаловаться на чужую, а также прочитать её описание, поставить лайк, оставить комментарий и перейти на страницу автора, нажав на его имя или аватар. На рисунках 26-27 показана страница публикации сайта Artfolio.

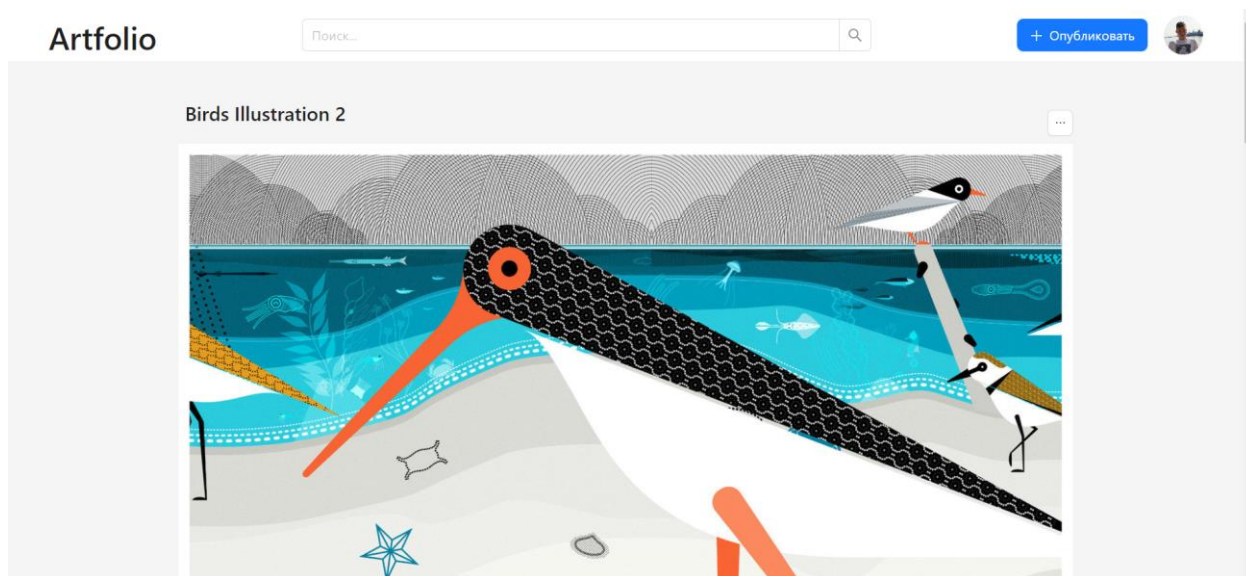


Рисунок 26 - Страница публикации

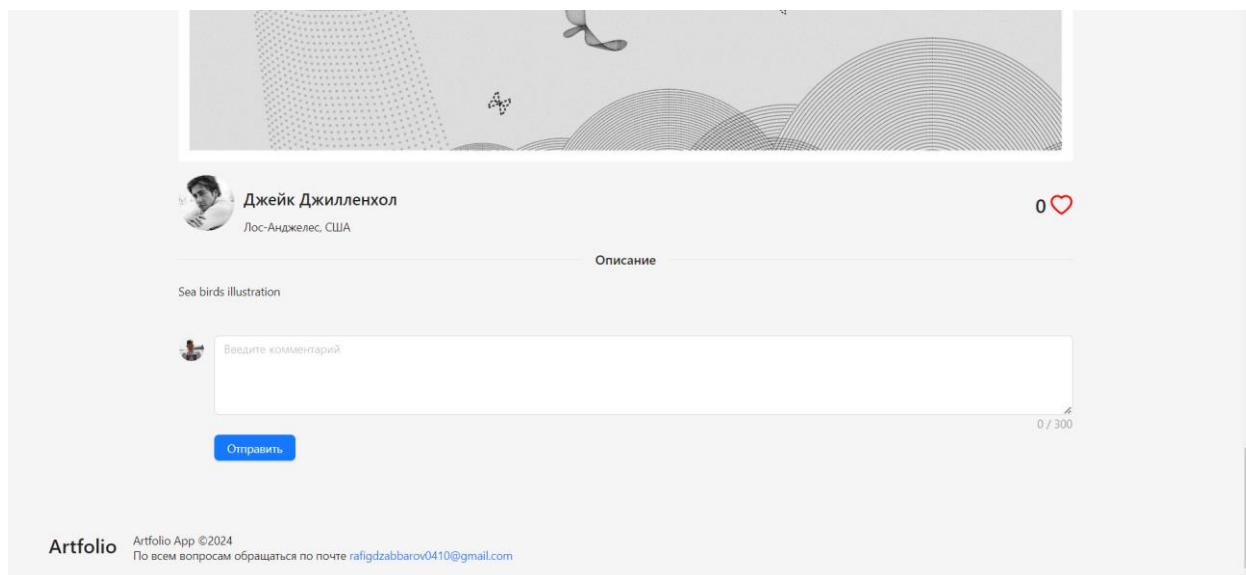


Рисунок 27 - Страница публикации

На странице профиля пользователь может подписаться на другого пользователя, связаться с ним, просмотреть его работы, а также просмотреть его статистику, описание профиля и экспортировать его через специальную кнопку. На рисунках 28-29 показана страница профиля пользователя.

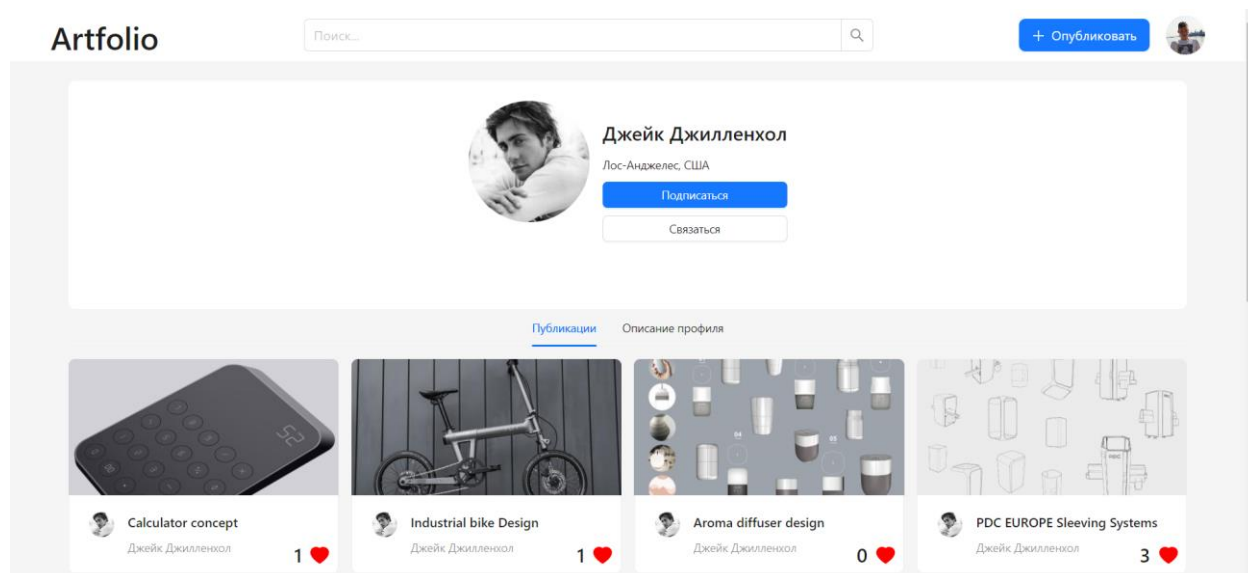


Рисунок 28 - Профиль пользователя с публикациями

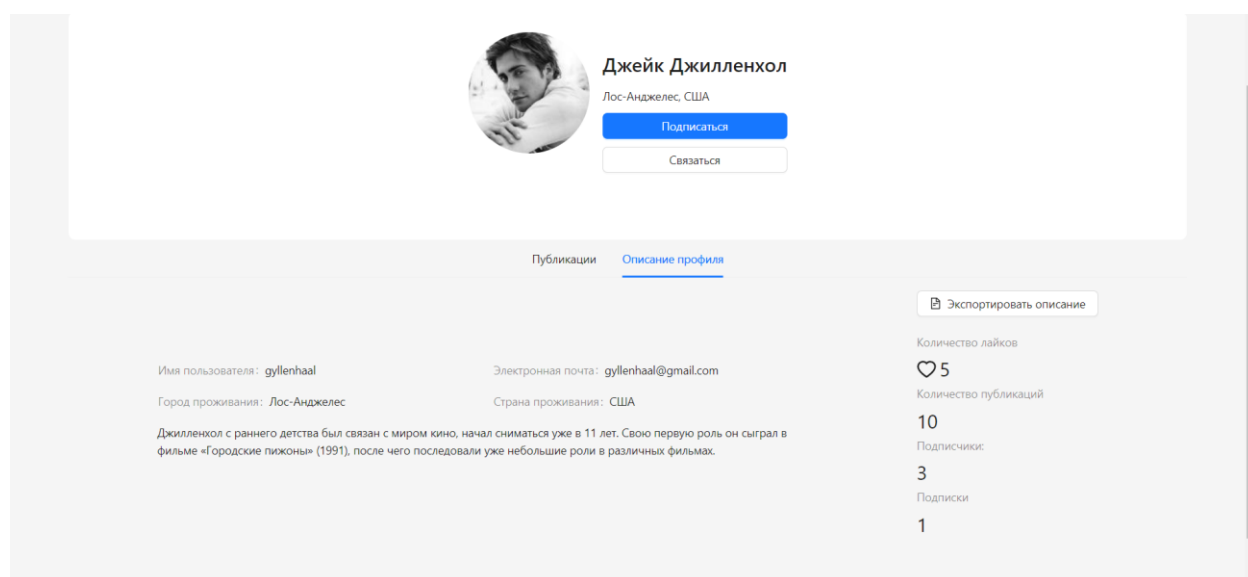


Рисунок 29 - Профиль пользователя с описанием, статистикой и кнопкой экспорта

Если пользователь не авторизован в системе, после нажатия на кнопку перед ним появится окно с выбором дальнейших действий, где пользователь

может выбрать вариант ввода логина и пароля для авторизации, восстановления пароля или регистрации. На рисунке 30 показана страница авторизации.

Artfolio

Поиск...

+ Опубликовать Войти

Войти в Artfolio

Электронная почта

Пароль

Забыли пароль

Войти

У вас нет аккаунта? Зарегистрироваться!

Artfolio Artfolio App © 2024  
По всем вопросам обращаться по почте rafgdtzabbarov0410@gmail.com

Рисунок 30 - Авторизация

В случае выбора регистрации, пользователь нажимает на текст «Зарегистрироваться», и перед ним открывается первая страница регистрации, после заполнения которой он переходит на вторую страницу регистрации. Если пользователь забыл пароль от аккаунта, есть возможность его восстановить при нажатии на кнопку «Восстановить пароль» на странице авторизации. На рисунках 31-32 показана страница регистрации.

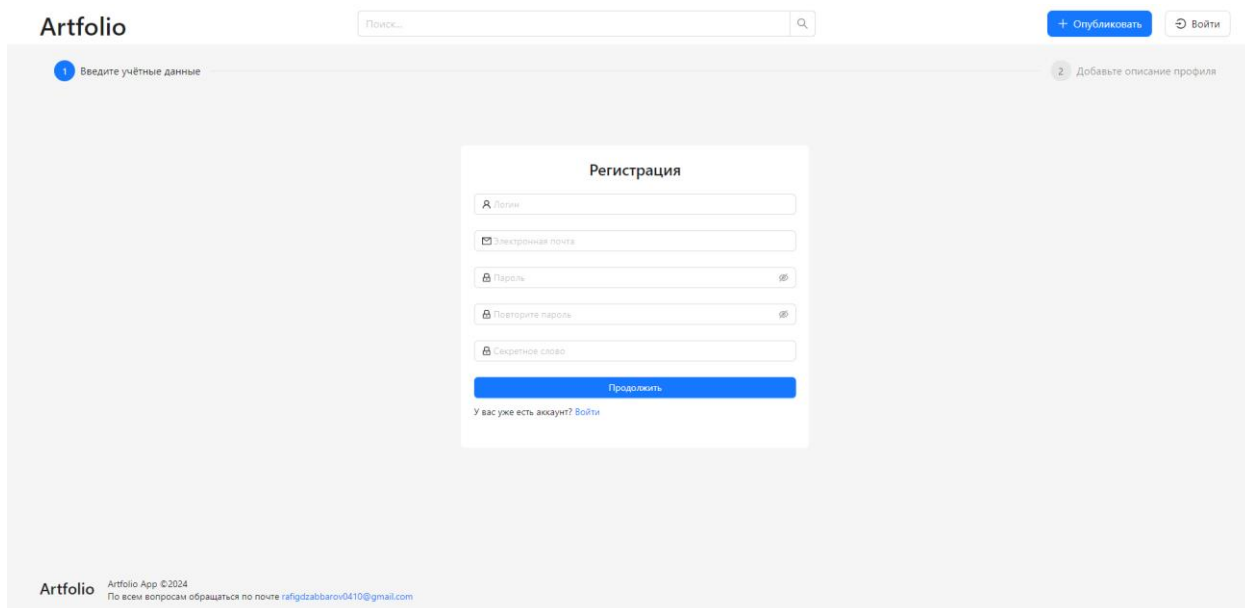


Рисунок 31 - Страница регистрации (1 часть)

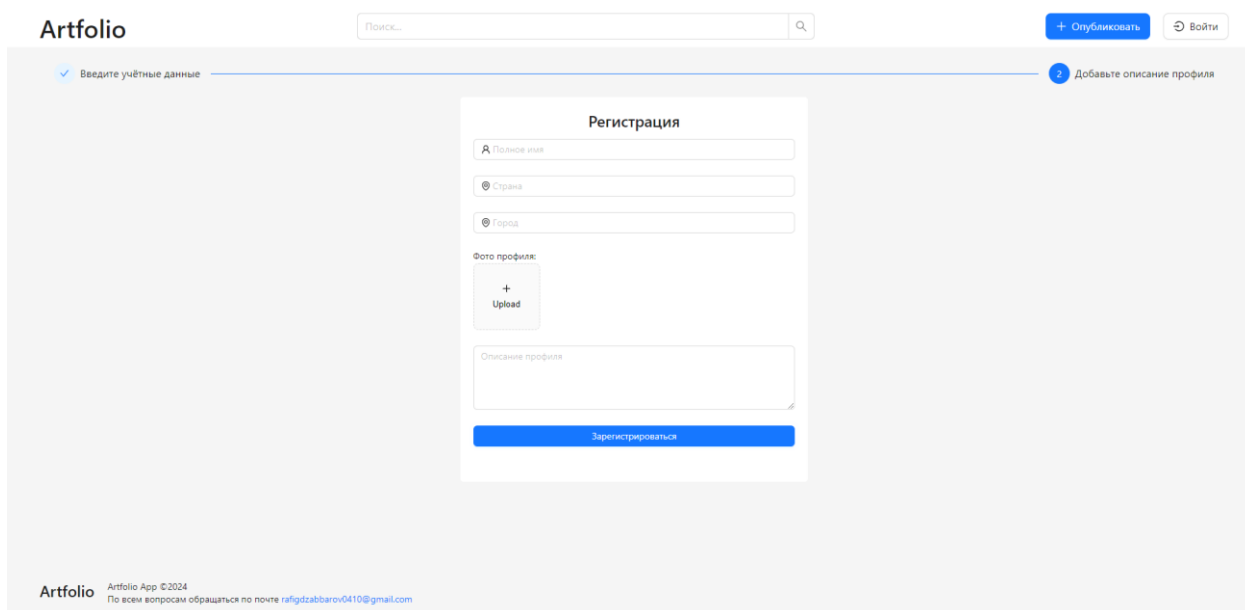


Рисунок 32 - Страница регистрации (2 часть)

Восстановление пароля проходит через ввод почты, секретного слова, задаваемого при регистрации, и нового пароля. На рисунке 33 показана страница восстановления пароля.

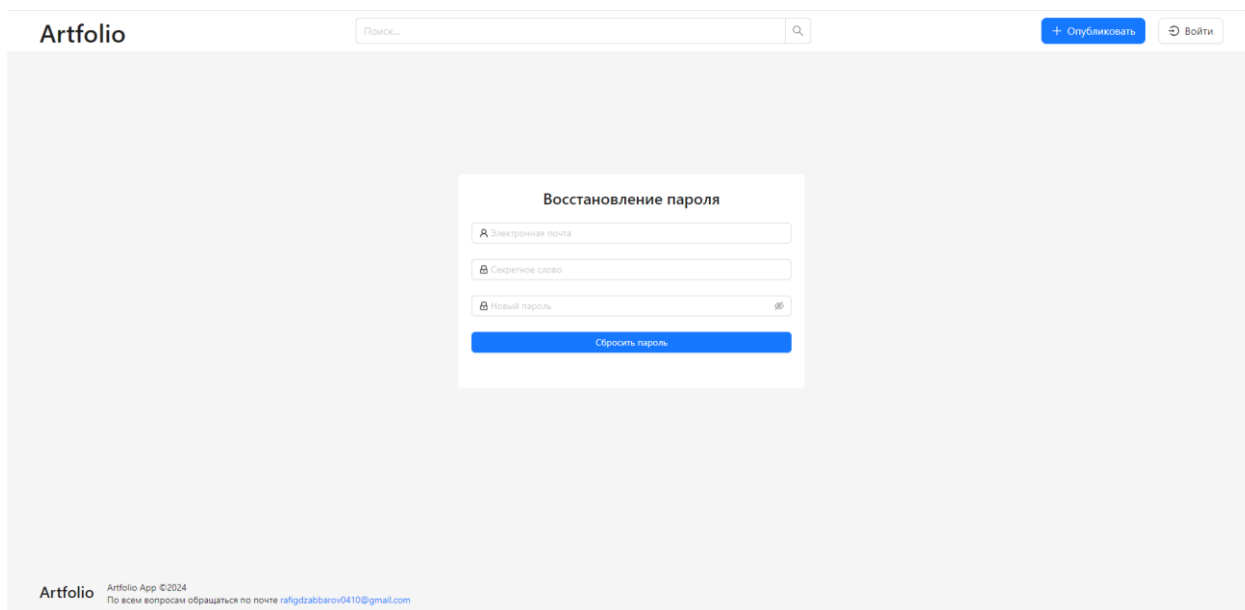


Рисунок 33 - Страница восстановления пароля с проверкой данных

После входа в систему пользователь сможет создать собственную публикацию через специальную кнопку в шапке страницы. Перед ним откроется страница, где он может ввести название публикации, её описание и загрузить фотографии, а также удалить их или поменять очередность. На рисунке 34 показана страница создания публикации.

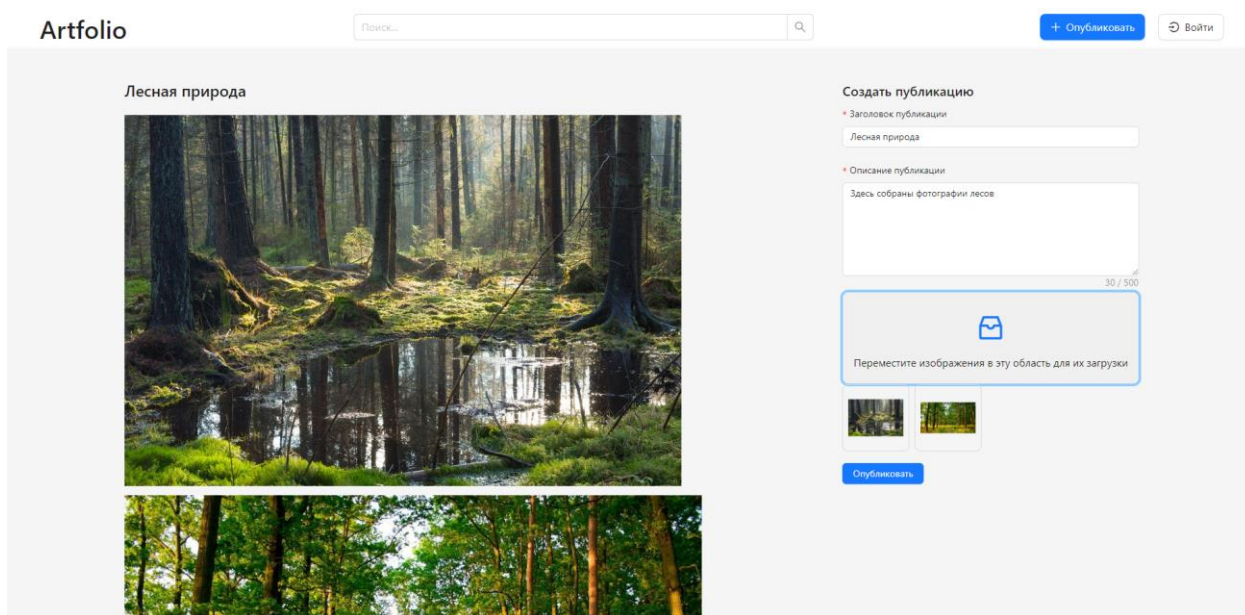


Рисунок 34 - Страница загрузки публикации.

## 4.4.2 Панель администратора

В возможности администратора входит рассмотрение жалоб на посты и комментарии. Это можно сделать на специальной странице «Панель администратора» в списке «Нерассмотренные». После рассмотрения жалобы администратором, она переносится в список «Рассмотренные». На рисунках 35-36 показана панель администратора.

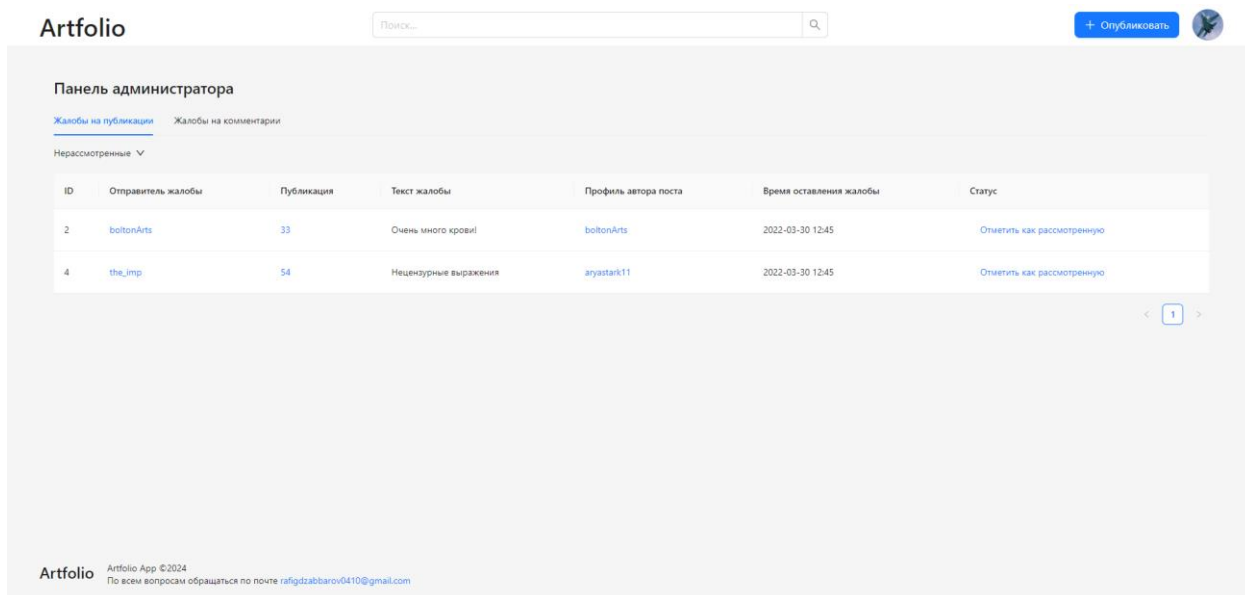


Рисунок 35 - Панель администратора

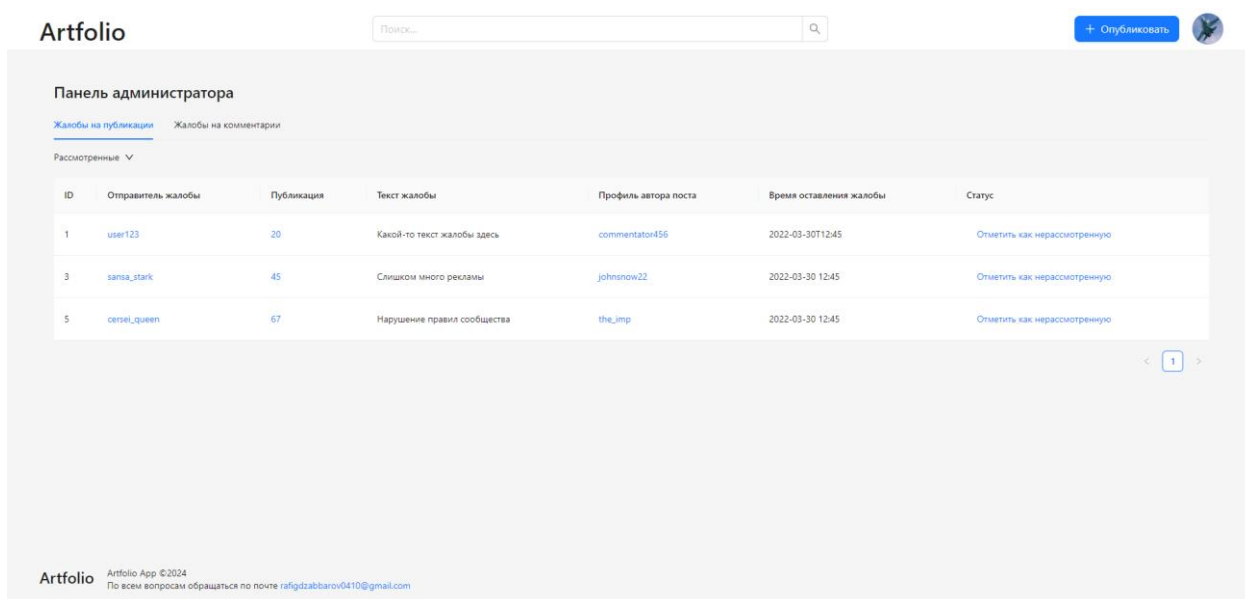


Рисунок 36 - Панель администратора

## 5 Тестирование

Для проверки работоспособности разработанного приложения было проведено тестирование системы с использованием следующего типа тестов:

- UI-тестирование (тестирование пользовательского интерфейса);
- Smoke testing (дымовое тестирование);

### 5.1 UI-тестирование

Тестирование пользовательского интерфейса (UI-тестирование) — это процесс проверки графического интерфейса приложения или веб-сайта на соответствие требованиям, удобство использования, визуальное оформление и правильное функционирование элементов пользовательского интерфейса.

Основная цель UI-тестирования заключается в убеждении, что пользовательский интерфейс выглядит и работает так, как ожидается, и что пользователи могут легко взаимодействовать с приложением или веб-сайтом.

В таблице 1 продемонстрированы результаты тестирования пользовательского интерфейса.

Таблица 1 - Результаты UI-тестирования

Тестовый сценарий	Ожидаемый результат	Статус теста
Нажатие на одну из категорий на главной странице	Переход на страницу товаров выбранной категории	Пройден
Нажатие на логотип	Переход на главную страницу	Пройден
Нажатие на кнопку «Авторизация» в шапке сайта	Переход на страницу авторизации	Пройден



Нажатие на кнопку «Войти» на странице авторизации с незаполненными полями	Вывод ошибки	Пройден
Нажатие на ссылку «Зарегистрироваться» на странице авторизации	Переход на страницу регистрации	Пройден
Нажатие на кнопку «Зарегистрироваться» на странице регистрации или на кнопку «Войти» на странице авторизации с незаполненными полями или неверной информацией	Вывод ошибки	Пройден
Нажатие на ссылку «Войти» на странице регистрации	Переход на страницу авторизации	Пройден
Нажатие на ссылку «Сбросить пароль» на странице авторизации	Переход на страницу сброса пароля	Пройден
Ввод названия публикации в строку поиска	Переход на страницу с найденными публикациями	Пройден
Нажатие на кнопку добавления публикации	Переход на страницу для создания публикаций	Пройден
Оставить пустыми поле названия или описания	Вывод ошибки	Пройден
Не загружать фотографии при создании публикации	Вывод ошибки	Пройден
Написание комментария на странице публикации	Публикация комментария	Пройден

Нажатие на кнопку «Лайк» на странице публикации	Увеличение количества лайков на 1	Пройден
Повторное нажатие на кнопку «Лайк» на странице публикации	Уменьшение количества лайков на 1	Пройден
Нажатие на кнопку «Пожаловаться»	Отображение окна для оставления жалобы	Пройден
Нажатие на кнопку «Отправить жалобу» без заполнения графов	Вывод ошибки	Пройден
Нажатие на кнопку «Редактировать профиль»	Переход на страницу редактирования профиля	Пройден
Заполнение графов при редактировании профиля	Обновление информации в профиле	Пройден
Допущение ошибок в заполнении графов при редактировании профиля	Вывод ошибки	Пройден
Нажатие на кнопку «Экспортировать описание» на странице описания профиля	Скачивание PDF-файла с описанием выбранного пользователя	Пройден
Открытие панели администратора	Переход на страницу с жалобами	Пройден
Нажатие на «Выйти»	Выход из аккаунта	Пройден

## 5.2 Дымовое тестирование

Дымовое тестирование (smoke testing) — это одна из методик функционального тестирования программного обеспечения, направленная на

быструю проверку основных функций или основных компонентов приложения с целью выявления критических проблем.

Целью дымового тестирования является убедиться, что основные функциональности приложения работоспособны, а другие компоненты системы могут быть интегрированы и протестированы более подробно в дальнейшем.

В таблице 2 продемонстрированы результаты дымового тестирования для сценариев пользователя и администратора.

Таблица 2 - Результаты дымового тестирования для пользователя

Тестовый сценарий	Результат теста
Регистрация	Пройден
Авторизация	Пройден
Сброс пароля	Пройден
Создание публикации	Пройден
Оценка публикации	Пройден
Отмена оценки публикации	Пройден
Редактирование данных профиля	Пройден
Выход из аккаунта	Пройден
Просмотр профиля	Пройден
Экспорт данных профиля	Пройден
Редактирование данных профиля	Пройден
Выход из аккаунта	Пройден

## 6 Анализ веб-приложения

Главной функциональностью приложения является создание публикаций и просмотр публикаций других авторов. Поэтому эта функциональность является важной для аналитики. С использованием сервиса Яндекс.Метрика были созданы цели и воронки для проведения анализа. На рисунке 37 показаны созданные цели и воронки.

Цели		
Цели позволяют отслеживать важные события на сайте: клики на кнопки, просмотры определенных страниц, скачивание файлов, отправку форм и многие другие. На одном счетчике может быть до 200 целей — автоцели не учитываются.		
<input checked="" type="checkbox"/> Разрешить в рекламных кампаниях оптимизацию по целям без доступа к счетчику		
<button>Добавить цель</button> <span>Избранные цели</span>		
Название цели	Описание	Номер цели
1   Регистрация на платформе	составная цель Зашло на страницу регистрации (ID: 330868748); • url содержит: /register Завершен первый этап регистрации (ID: 330868749); • событие идентификатор цели: step1Success Завершен второй этап регистрации (ID: 330868750); • событие идентификатор цели: step2Success	330868741
2   Переход на страницу профиля	url содержит: /profile	330868377
3   Клик по кнопке "Связаться" в профиле	идентификатор: contactTo	330868463
4   Экспорт описания профиля к себе	идентификатор: exportTo	330868464
5   Открытие страницы публикации	url содержит: /posts/{id}	330868574
6   Создание публикации	составная цель Переход на страницу создания публикации (ID: 330868751); • url содержит: /posts Завершение создания публикации (ID: 330868752); • событие идентификатор цели: createSuccess	330868698

Рисунок 37 - Созданные цели на сервисе Яндекс.Метрика

Была добавлена продуктовая воронка по созданию публикации (Рисунок 38). Она демонстрирует заинтересованность пользователей в возможности представления своих работ.

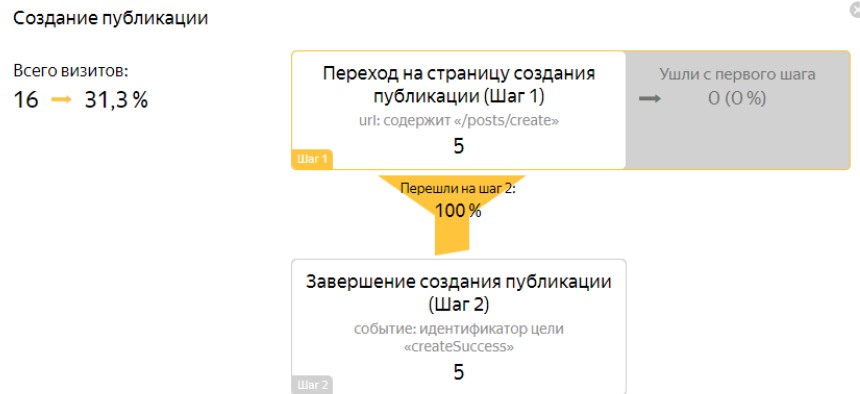


Рисунок 38 - Воронка «Создание публикации»

Данная воронка отражает количество пользователей, зашедших на страницу создания публикации, а именно 16. К шагу завершения создания публикации перешли 16 человек, то есть 100% от тех, кто перешел на страницу создания.

## **ЗАКЛЮЧЕНИЕ**

В ходе данной курсовой работы были выполнены все поставленные задачи, связанные с разработкой онлайн-платформы для создания и управления портфолио для художников, дизайнеров и фотографов. Была реализована функциональность создания публикации и просмотра публикаций других пользователей. Также была добавлена возможность связаться с заинтересовавшим пользователем и экспортировать описание его профиля себе.

Для разработки платформы использовался ряд современных технологий, что позволило реализовать удобный и функциональный веб-ресурс. Этот проект считается выполненным успешно и полностью соответствует поставленной цели.

В заключение следует отметить, что разработанный функциональный прототип онлайн-платформы для управления портфолио предоставляет пользователям удобный инструмент для создания, редактирования и демонстрации своих творческих работ. Дальнейшее развитие и расширение функциональности платформы планируется на основе обратной связи от пользователей и требований рынка.